陈恒 刘鑫 陶永鹏◎编著

HTML与CSS

教学做一体化教程

网页设计

清华大学出版社

HTML与CSS网页设计教学做一体化教程

陈　恒　刘　鑫　陶永鹏　编著

清华大学出版社
北　京

内 容 简 介

HTML 与 CSS 网页设计是动漫设计与制作、视觉效果设计、多媒体技术、计算机网络技术等专业的必修课程,也是许多计算机爱好者所希望掌握的技能。

本教材采用教、学、做一体化模式,以核心知识、能力目标、任务驱动和实践环节为单元组织本教材的体系结构。精选大量的真实案例,循序渐进地介绍了使用 HTML 和 CSS 创建网页以及进行页面布局的基础知识。注重结合实例讲解一些难点和关键技术,在实例上侧重实用性和启发性。特别强调知识点的能力目标,通过合理的任务驱动和实践环节提高静态网页制作的能力。全书分为 12 章,内容包括:初始 HTML、HTML 文件基本标记、文字布局与文字效果、列表、图像与多媒体、超链接、表格、表单与页面控件、框架与浮动窗口、JavaScript、CSS、地址簿网站的布局。

本书不仅适合作为高等院校动漫设计与制作、计算机网络技术等专业的相关课程教材,而且特别适合于作为"教、学、做"一体化的教材。

图书在版编目(CIP)数据

HTML 与 CSS 网页设计教学做一体化教程/陈恒,刘鑫,陶永鹏编著. —北京:清华大学出版社,2013.4
ISBN 978-7-302-31062-4

Ⅰ. ①H…　Ⅱ. ①陈… ②刘… ③陶…　Ⅲ. ①超文本标记语言－程序设计 ②网页制作工具　Ⅳ. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字(2012)第 303482 号

责任编辑: 田在儒
封面设计: 李　丹
责任校对: 刘　静
责任印制:

出版发行: 清华大学出版社
网　　址: http://www.tup.com.cn, http://www.wqbook.com
地　　址: 北京清华大学学研大厦 A 座　　**邮　　编:** 100084
社 总 机: 010-62770175　　**邮　　购:** 010-62786544
投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载: http://www.tup.com.cn,010-62795764
印 刷 者:
装 订 者:
经　　销: 全国新华书店
开　　本: 185mm×260mm　　**印　　张:** 14　　**字　　数:** 334 千字
版　　次: 2013 年 4 月第 1 版　　**印　　次:** 2013 年 4 月第1 次印刷
印　　数: 1～0000
定　　价: .00 元

产品编号:045585-01



本教材采用教学做一体化的方式撰写，合理地组织学习单元，并将每个单元分解为核心知识、能力目标、任务驱动、实践环节4个模块，体现教学做一体化过程。

每个单元的核心知识点强调在静态网页制作中最重要和实用的知识，能力目标强调使用核心知识点进行静态网页制作的能力，需要完成的任务中的“任务模板”起着训练静态网页制作能力的作用，其中的任务总结主要总结任务中涉及的重要技巧、注意事项以及扩展知识，通过该“任务模板”的训练，读者有能力完成后续的实践环节。第1章主要介绍HTML文件的基本结构以及如何运行HTML文件。第2章是HTML文件基本标记，在核心知识点和任务的安排方面特别注重训练初学者应当掌握和理解重要的基础知识以及知识点的能力目标。第3章是文字布局与文字效果，在任务安排上注重结合实际问题训练读者熟练地设计网页中的文字布局与文字效果。第4章是列表，着重讲解在网页中使用列表设计结构化的、易读的文本。第5章是图像与多媒体，在任务安排上注重结合实际问题训练读者熟练地使用图像与多媒体美化网页。第6章是超链接，主要讲述了在网页中如何使用超链接。第7章是表格，重点介绍了表格的创建、属性以及结构。第8章是本书的重点内容之一，主要讲述在网页中使用表单实现程序与用户的交互。第9章介绍了框架与浮动窗口的用法。第10章是JavaScript，重点介绍了JavaScript的基本语法，在任务安排上注重结合实际问题训练读者在网页中熟练地使用JavaScript完成客户端的特定功能。第11章是本书的重点内容之一，主要介绍了CSS的基本语法、经典的网页布局。第12章是一个综合网站案例，通过一个小型的地址簿网站，讲述如何采用DIV+CSS进行网站的前台布局设计。

本教材特别注重引导学生参与课堂教学活动，适合高等院校相关专业作为教、学、做一体化的教材。

本教材的示例和任务模板的源程序以及电子教案可以在清华大学出版社网站免费下载，以供读者和教学使用。

由于编者水平有限，书中难免存在错误和疏漏之处，敬请广大读者给予批评指正。

编　者

2013年3月

目　录
CONTENTS

第 1 章

初始 HTML

主要内容

- HTML 文件的基本结构
- 简单的 HTML 页面

可视化的 HTML 开发工具固然方便快捷，如 FrontPage、Dreamweaver 等，但是，可视化开发工具自身也会产生大量不必要的 HTML 代码，这样会为以后的程序维护埋下隐患。一个优秀的网页设计者应该在掌握 HTML 语言的基础上，进一步熟练使用可视化开发工具，从而达到快速制作网页和优化网页的目的。

本章通过简单的实例掌握 HTML 文件的基本结构和 HTML 程序的开发步骤。

1.1 HTML 文件的基本结构

1.1.1 核心知识

HTML 的英文全称是 Hyper Text Markup Language，即超文本标记语言，它是用于编写网页的主要标记语言。下面是一个 HTML 文件的基本结构。

```
<html>
    <head>
        …
    </head>
    <body>
        …
    </body>
</html>
```

从上面的代码段可以看出，HTML 文件的基本结构分为 3 部分，其中各部分含义见下文所述。

<html>…</html>：表示 HTML 文件开始和结束的位置，里面包括<head>和<body>等标记。HTML 文件中所有的内容都应该在这两个标记之间。

<head>…</head>：HTML 文件的头部标记，习惯将这两个标记之间的内容统称为 HTML 的头部。

<body>...</body>：用来指明文档的主体区域，网页所要显示的内容都要放置在这个标记内。习惯将这两个标记之间的内容统称为 HTML 的主体。

1.1.2 能力目标

了解 HTML 的基本概念，认识并理解 HTML 文件的基本结构。

1.1.3 任务驱动

1. 任务的主要内容

把“初次见面，请多多照顾”这句话显示在网页的主体区域中。

2. 任务的代码模板

将下列 example1_1.html 中的【代码】替换为 HTML 代码，有关代码要求：【代码 1】为 title 标记的结束，【代码 2】为 body 标记的开始，【代码 3】为网页主体的内容“初次见面，请多多照顾”。

example1_1.html

```
<html>
    <head>
        <title>第一个 HTML 页面【代码 1】
    </head>
    【代码 2】
        【代码 3】
    </body>
</html>
```

3. 任务小结或知识扩展

HTML 文件中的标记应该有开始标记和结束标记，例如，以<title>开始，以</title>结束。每个文件都有自己的扩展名，HTML 文件的扩展名为.html 或.htm。

4. 代码模板的参考答案

```
【代码 1】：</title>
【代码 2】：<body>
【代码 3】：初次见面，请多多照顾
```

1.1.4 实践环节

- 把 example1_1.html 完整的代码保存在操作系统自带的记事本(文本文档)中。选择记事本菜单中的“文件”→“另存为”命令，在打开的对话框中选择保存的路径，在“文件名”文本框中输入“example1_1.html”，单击“保存”按钮，如图 1.1 所示。
- 双击“example1_1.html”文件，观察运行效果。



图 1.1　HTML 程序的保存

1.2　使用记事本编写 HTML 页面

1.2.1　核心知识

编写 HTML 页面有两种常用方法：一种是利用操作系统自带的记事本编写；另一种是利用可视化网页制作软件(如 Dreamweaver)编写。使用可视化网页制作软件编写网页在本书中不涉及。下面讲述使用记事本手工编写 HTML 页面。

HTML 语言并不需要什么特殊的开发环境，可以直接在操作系统自带的记事本中编写。使用记事本手工编写 HTML 页面的具体操作步骤如下。

(1) 在 Windows 操作系统下，选择“开始”→“程序”→“附件”→“记事本”命令，新建一个记事本，在记事本中输入 HTML 代码。

(2) 编写完 HTML 文件后，选择“文件”→“另存为”命令，在“另存为”对话框中选择保存的路径，在“文件名”文本框中输入文件的名字，如 myPage.html，文件的扩展名为.html 或.htm。

(3) 单击“保存”按钮，这时该文本文件就保存成了 HTML 文件，双击 HTML 文件就可以看到页面运行的效果。

1.2.2　能力目标

使用记事本手工编写简单的 HTML 页面。

1.2.3　任务驱动

编写一个简单的 HTML 页面，该页面的中央位置显示两行文字：“很高兴学习 HTML 语言”和“我们以后要好好学习 HTML 语言”。页面运行效果如图 1.2 所示。任务的主要步骤如下。

- 使用记事本编写、保存 HTML 页面。
- 使用浏览器运行 HTML 页面。



图 1.2 example1_2.html 页面运行效果

1. 使用记事本编辑器编写、保存 HTML 文件

(1) 输入文件 example1_2.html 的内容。

在打开的记事本编辑器中输入如下内容：

```
<html>
    <head>
        <title>我的 HTML 页面</title>
    </head>
<body>
    <center>
        很高兴学习 HTML 语言<br>
        我们以后要好好学习 HTML 语言
    </center>
</body>
</html>
```

(2) 保存 example1_2.html 文件。

参照 1.1 节实践环节或本节的核心知识，保存 example1_2.html 文件。

2. 使用浏览器运行 HTML 文件

双击“example1_2.html”文件的图标，即可打开该文件并可看到网页的运行效果。HTML 文件的图标如图 1.3 所示。



图 1.3 HTML 文件的图标

3. 查看 HTML 页面的源代码

对于学习 HTML 网页而言，可以查看其他人的网页代码是一件很美妙的事情。只需要简单操作就可以查看别人的网页是如何构建的，能够学到很多知识。但是如何查看呢？

要想查看 HTML 页面的源代码，就是查看它的源文件，在 IE 浏览器中查看网页的源代码有以下两种常用方法。

- 把光标移到网页中的某个位置(别放在图像上)，然后右击，在弹出的快捷菜单中选择“查看源文件”命令，就可以查看源文件。
- 在浏览器菜单栏上，选择菜单中的“查看”→“查看源文件”命令，也可以查看源文件。

4. 任务小结或知识扩展

在编写 HTML 页面时，要注意以下几点。

（1）“<”和“>”是任何标记的开始和结束。标记要用尖括号括起来，并且在结束标记的前面加一个“/”，如<body></body>。

（2）在源代码中任何回车和空格都不影响网页的运行效果，但为了程序清晰易读，建议不同的标记之间用回车进行换行。

（3）在源代码中不区分大小写。

（4）要正确输入标记。输入标记时，不要输入多余的空格，否则浏览器可能无法识别这个标记，导致无法正确显示信息。

1.2.4　实践环节

编写一个网页，在网页中央位置显示“关关雎鸠，在河之洲。窈窕淑女，君子好逑。”网页运行效果如图 1.4 所示。并在 IE 中查看该网页的源代码。



图 1.4　关雎《诗经》的运行效果

1.3　使用 Eclipse 编写 HTML 页面

HTML 页面除了利用操作系统自带的记事本和可视化网页制作软件编写之外，还可以使用 IDE(集成开发环境)工具编写。

1.3.1　核心知识

本书中使用的 IDE 工具是 Eclipse。Eclipse 是一个可用于开发 Java 程序的 IDE 工具。登录 http://www.eclipse.org，在 Downloads 里选择 Eclipse IDE for Java EE Developers，然后在 Download Links 里选择适用 Windows 32-bit 的 Eclipse。如果读者使用 64 位机或其他的操作系统，可以在下载列表中选择下载相应的 Eclipse。

为了保证 Eclipse 的正常运行，应事先为其安装 JRE(Java 运行环境)。可以登录 http://www.oracle.com/technetwork/java，在 Software Downloads 里选择 Java SE 提供的 JRE。对于 Windows 操作系统，选择下载 jre-7u5-windows-i586.exe，如果使用 64 位机或其他的操作系统，可以在下载列表中选择下载相应的 JRE。JRE 下载完成后，按默认安装即可完成安装。

Eclipse 下载完成后，解压到自己设置的路径下，即可完成安装。Eclipse 安装完成后就

可以启动了。双击 Eclipse 安装目录下的 eclipse.exe 文件，启动 Eclipse。在初次启动时，需要设置工作空间，比如将工作空间设置为 E:\eclipse\workspace，如图 1.5 所示。

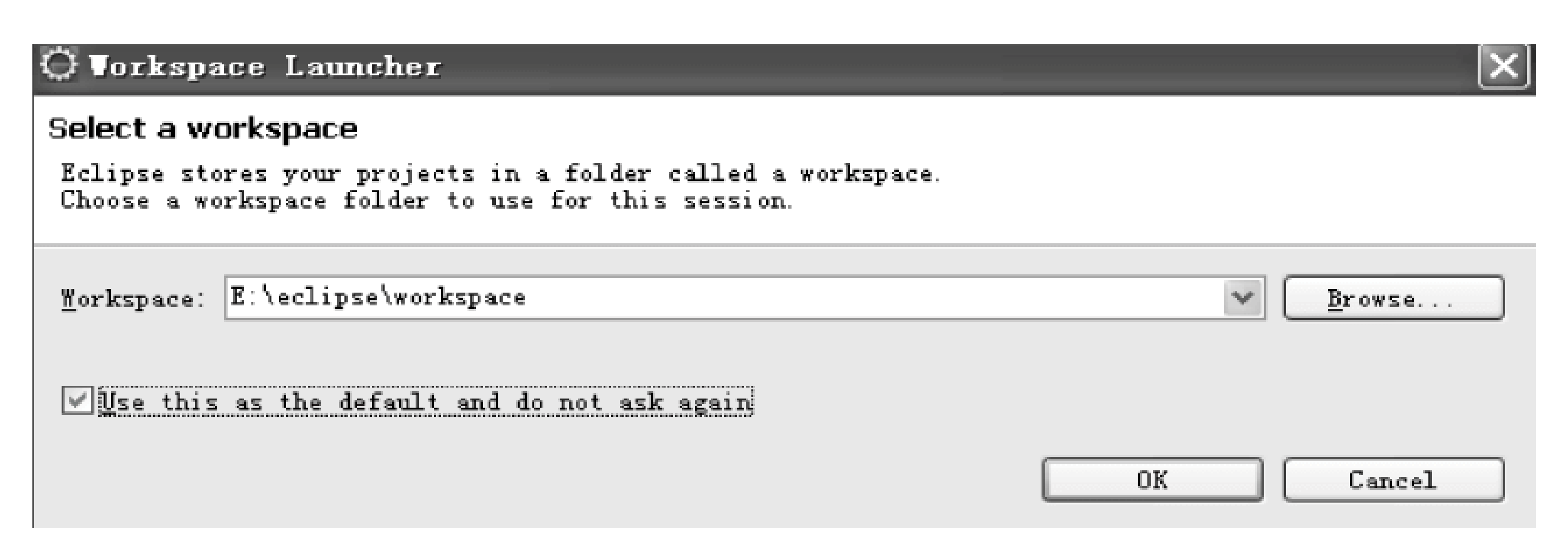


图 1.5　设置工作空间

在图 1.5 所示的界面上单击 OK 按钮进入如图 1.6 所示的欢迎页面。

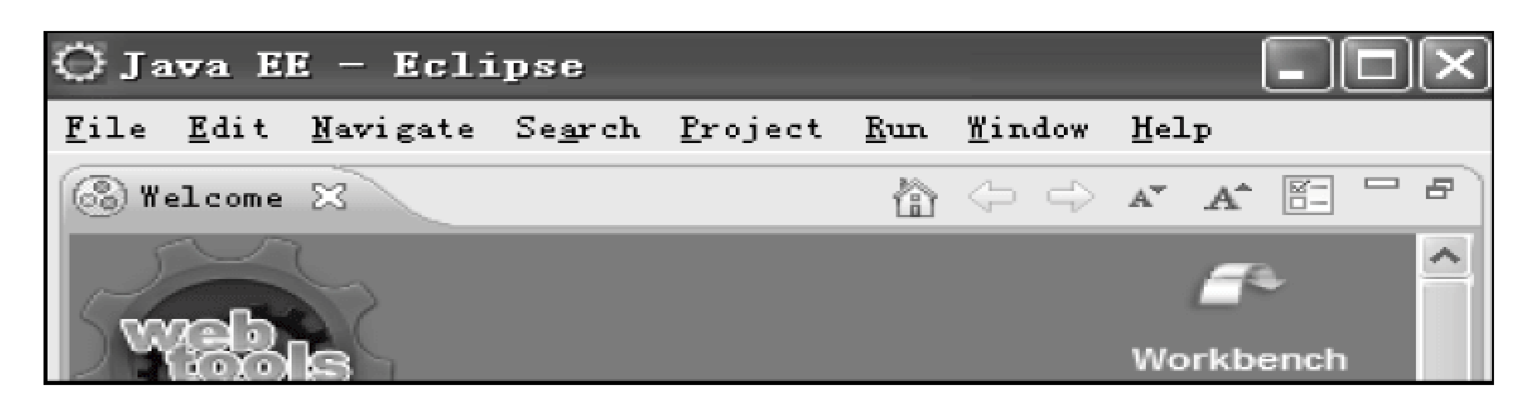


图 1.6　Eclipse 欢迎页面

1.3.2　能力目标

使用 Eclipse 编写简单的 HTML 页面。

1.3.3　任务驱动

1. 任务的主要内容

(1) 创建项目

① 启动 Eclipse，进入 Eclipse 的开发界面。

② 选择主菜单中的 File→New→Project 命令，打开 New Project 对话框，在该对话框中选择 Web 节点下的 Static Web Project 子节点，如图 1.7 所示。

③ 单击 Next 按钮，打开 New Static Web Project 对话框，在该对话框的 Project Name 文本框中输入项目名称，这里为 htmlFirst。单击 Finish 按钮，完成项目 htmlFirst 的创建。此时在 Eclipse 平台左侧的 Project Explorer 选项卡中，将显示项目 htmlFirst，依次展开各节点，可显示如图 1.8 所示的目录结构。

(2) 创建 HTML 页面

htmlFirst 项目创建完成后，可以根据实际需要创建 HTML 文件。下面将创建一个名字为 myFirst.html 的 HTML 文件。

① 选中 htmlFirst 项目的 WebContent 节点并右击，在弹出的快捷菜单中，选择 New→HTML File 命令，打开 New HTML File 对话框，在该对话框的 File name 文本框中输入文件名 myFirst.html，其他采用默认设置，单击 Finish 按钮完成 HTML 文件的创建。

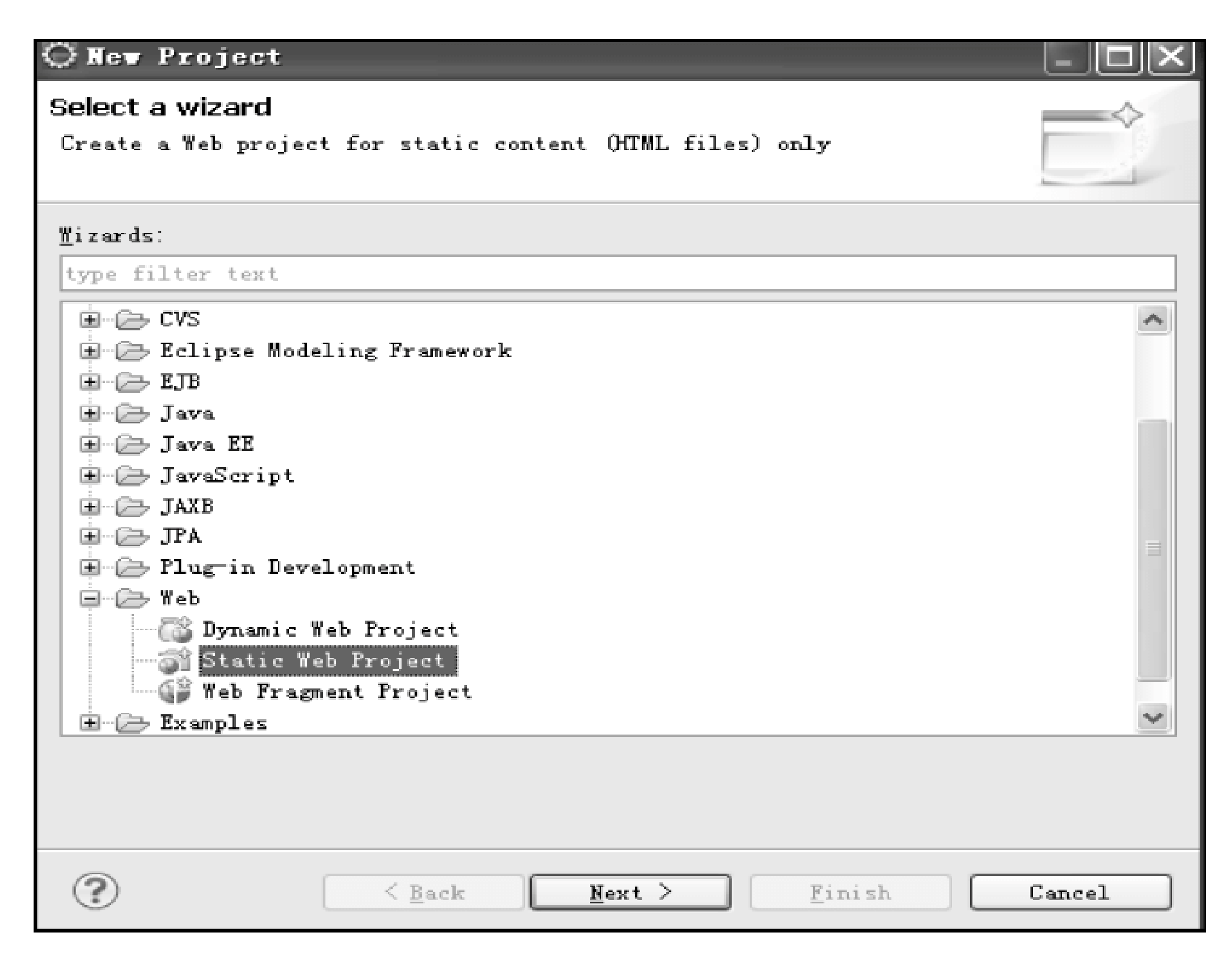


图 1.7　New Project 对话框

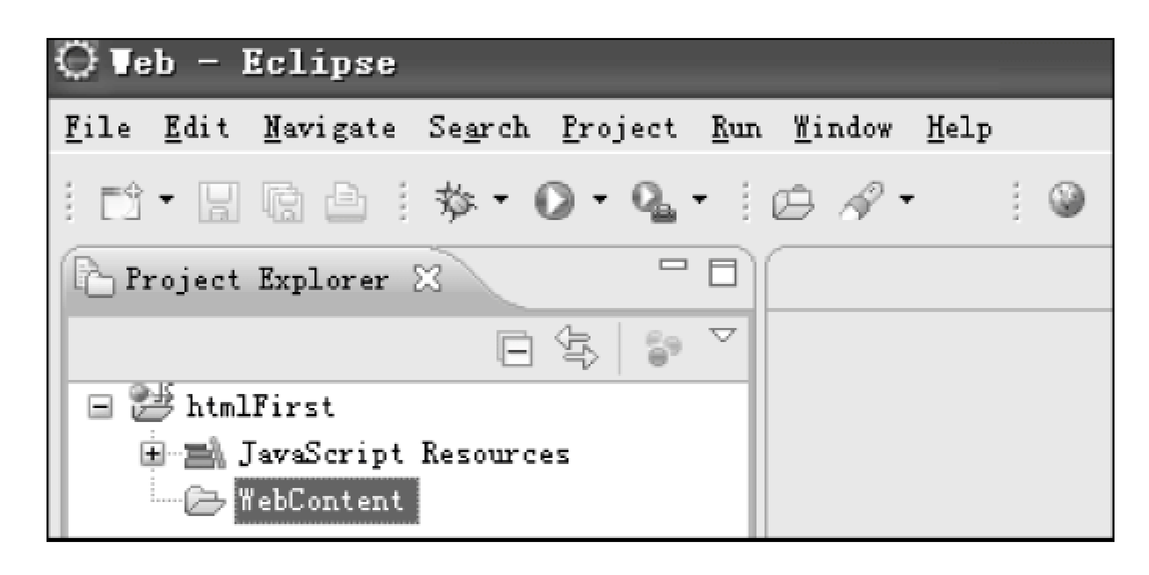


图 1.8　项目 htmlFirst 的目录结构

② HTML 文件创建完成后，在 htmlFirst 项目的 WebContent 节点下，自动添加一个名称为 myFirst.html 的 HTML 文件，同时，Eclipse 会自动将 HTML 文件在右侧的编辑框中打开。

③ 将 myFirst.html 文件中的默认代码修改为以下代码：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <center>真高兴,忙活半天了,终于要看到人生中第一个 HTML 页面了。</center>
</body>
</html>
```

④ 将编辑好的 HTML 页面保存(组合键 Ctrl+S)，至此完成了一个简单的 HTML 页面的创建。

(3) 运行 HTML 页面

完成 HTML 页面的创建后，运行该页面。下面介绍具体的方法。

① 在 htmlFirst 项目的 WebContent 节点下，找到 myFirst.html 选中该 HTML 文件并右击，在弹出的快捷菜单中，选择 Run As→Run On Server 命令，打开 Run On Server 对话框，在该对话框中，选中 Basic 中的 J2EE Preview，其他采用默认设置，如图 1.9 所示。



图 1.9　Run On Server 对话框

② 单击 Finish 按钮，即可运行该 HTML 页面，运行后的效果如图 1.10 所示。如果想在 IE 浏览器中运行该页面，可以将图 1.10 中的 URL 地址复制到 IE 浏览器的地址栏中，并按 Enter 键运行即可。



真高兴，忙活半天了，终于要看到人生中第一个HTML页面了。

图 1.10　运行 HTML 页面

2. 任务小结或知识扩展

在一个 Static Web Project(静态项目)的 WebContent 节点下可以创建多个 HTML 文件，另外 HTML 文件中使用到的图片文件、音乐文件、CSS 文件(层叠样式表)以及 JavaScript 文件都放在 WebContent 节点下。有关 CSS 和 JavaScript 在 HTML 页面中的应用，在本书后续的内容中将介绍。

另外，请注意 HTML 页面运行时并不需要什么特殊环境，使用 Eclipse 编写并运行只是为了方便而已。但有时候 Eclipse 自带的 J2EE Preview 不能运行个别 HTML 页面(如下载页面)，这时就需要使用 IE 浏览器运行了。

1.3.4 实践环节

参照本节的任务内容，完成以下实践内容。

• 在 htmlFirst 项目中创建一个名称为 yourFirst.html 文件，在 HTML 页面中显示“不错！不错！自己能创建 HTML 页面，并且可以运行了”。
• 使用 Eclipse 运行 yourFirst.html 页面。
• 使用 IE 浏览器运行 yourFirst.html 页面。

1.4　小　　结

HTML 是以文字为基础的语言，并不需要什么特定的开发环境，可以直接在 Windows 自带的记事本中编写。HTML 程序，也不需要什么特定的运行环境，可以使用 Windows 自带的 IE 运行 HTML 程序。

使用 Eclipse 集成开发工具编写并运行 HTML 程序，只是为了方便而已，并不是说 HTML 程序需要 Eclipse 才能编写与运行。

习　题　1

1. 为了标识一个 HTML 文件应该使用的 HTML 标记是(　　)。
   A. <p></p>　　B. <boby></body>
   C. <html></html>　　D. <table></table>
2. 可以不用发布到 Web 服务器，就能在本地计算机上浏览的网页编写语言是(　　)。
   A. ASP　　B. HTML　　C. PHP　　D. JSP
3. 用 HTML 标记语言编写一个简单的网页，网页最基本的结构是(　　)。
   A. <html> <head>…</head> <frame>…</frame> </html>
   B. <html> <title>…</title> <body>…</body> </html>
   C. <html> <title>…</title> <frame>…</frame> </html>
   D. <html> <head>…</head> <body>…</body> </html>
4. 下列对 HTML 的含义解释正确的是(　　)。
   A. HTML 即 Hyper Text Markup Language 超文本标记语言
   B. HTML 是编程语言
   C. HTML 是一种开发工具
   D. HTML 是一种浏览器
5. 下列关于 HTML 文件的说法错误的是(　　)。
   A. HTML 文件是一个包含标记的文本文件
   B. 这些标记告诉浏览器怎样显示这个页面
   C. HTML 文件必须以.htm 为扩展名
   D. HTML 文件可以用一个简单的文本编辑器创建
6. 简述 HTML 文件的基本结构。
7. 简述使用记事本编辑器编写 HTML 页面的操作步骤。
8. 简述编写 HTML 页面的注意事项。

第 2 章

HTML 文件基本标记



- head 和 title 标记
- 元信息标记 meta
- 主体标记 body

一个完整的网页应该包含头部和主体两部分。头部，主要包括当前 HTML 文件的相关信息，如标题、编码方式和关键字等。主体，主要包括页面的主体内容，如文字、图像和表格等。本章将介绍这些基本标记，它们都是一个完整的网页必不可少的标记。

2.1　head 和 title 标记

2.1.1　核心知识

在 HTML 文件的头部标记中，一般需要包括标题和元信息。头部标记包含的是与页面相关的信息，而不是页面本身所展现的内容。如它包含的 title 标记，就是用来告诉浏览器在页面标题栏上显示什么内容。

1. head 标记

head 标记的格式如下：

```
<head> ...</head>
```

定义在 HTML 文件头部的内容不会直接显示在网页上。

2. title 标记

title 标记的格式如下：

```
<title> ...</title>
```

<title>和</title>标记位于 HTML 文件的头部，即位于<head>和</head>标记之间。<title>和</title>标记中间的内容就是标题内容，它可以帮助用户更方便地识别页面。

2.1.2　能力目标

能够使用 title 标记设置网页的标题。

2.1.3　任务驱动

1. 任务的主要内容

编写一个 HTML 页面，该页面的标题为“学习 head 和 title 标记”。网页显示的内容为：“设置 HTML 页面的标题时应尽可能准确地描述页面的内容，表达清晰。”

2. 任务的代码模板

将下列 example2_1.html 中的【代码】替换为 HTML 代码。

example2_1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
        【代码 1】<!-- 设置 title 标记 -->
        【代码 2】<!-- 结束 head 标记 -->
        【代码 3】<!-- body 标记开始 -->
        <center>
            设置 HTML 页面的标题时应尽可能准确地描述页面的内容,表达清晰。
        </center>
    </body>
</html>
```

3. 任务小结或知识扩展

title 标记的内容还有如下一些其他用途。

- 对于打开的文档，在 Windows 任务栏中都会显示出来。在打开多个窗口时，就可以使用标题识别不同的文档。
- 当搜索引擎搜索页面时，一定程度上依赖于标题的内容，而且这些内容也将显示在搜索结果中。

4. 代码模板的参考答案

```
【代码 1】: <title>学习 head 和 title 标记</title>
【代码 2】: </head>
【代码 3】: <body>
```

2.1.4　实践环节

- 将任务中 example2_1.html 的代码使用浏览器运行一下，看看网页标题显示的位置及效果。
- 将 example2_1.html 网页中的标题改成“今天天气真好，我们去看海吧！”，再看看网页的运行效果。

2.2 使用meta标记定义编码方式、定时跳转以及过渡效果

2.2.1 核心知识

在head标记里还可以有任意数量的meta标记，它的主要功能是定义页面的信息。这些信息并不显示在浏览器中，例如，需要向浏览器或搜索引擎提供页面作者的名字、关键字及描述信息等。

1. 定义页面关键字

定义页面关键字的格式如下：

```
<meta name="keywords" content="关键字内容" >
```

其中，name为属性名称，这里是keywords代表设置网页的关键字，在content属性中则定义具体的关键字内容。关键字可以提高被搜索引擎搜索到的概率，多个关键字时，关键字之间使用逗号隔开。

2. 定义网页语言的编码方式

meta标记可以用来告诉浏览器使用什么字符集显示网页。这样浏览器就可以正确地选择语言，而不需要人工选择。meta标记的格式如下：

```
<meta http-equiv="Content-Type" content="text/html;charset=字符集类型">
```

其中，http-equiv表示传送HTTP通信协议的头，而在content中是具体的属性值。每个属性都是由name和value两部分组成的，其格式总是name=value。charset用于设置网页的字符编码类型，简体中文设置为gb2312，英文设置为ISO-8859-1，此外，还有其他的字符集，如utf-8(网页不管使用什么语言都适用)。

3. 定义网页的定时跳转

有些情况下，一个网页需要自动跳转到其他网页中，这就是网页的定时跳转。设置网页定时跳转的格式如下：

```
<meta http-equiv="refresh" content="跳转时间;URL=跳转的页面地址">
```

其中，refresh设置网页的刷新，content中给出了刷新的时间和刷新后的页面地址，时间和地址之间用分号隔开。默认情况下，跳转时间以秒为单位。

4. 定义网页的过渡效果

当进入或离开页面时，页面具有不同的切换效果，这就是页面的过渡效果。添加此功能可以使网页看起来更具动感，但有的IE不支持过渡效果。设置网页过渡效果的格式如下：

```
<meta http-equiv="过渡事件" content="revealtrans(duration=过渡效果持续时间,
transition=过渡方式)">
```

其中，过渡事件可以是进入页面或离开页面。进入页面事件，http-equiv值为page-enter;离开页面事件，http-equiv值为page-exist。过渡效果持续时间默认以秒为单位，过渡方式的编码如表2.1所示。

表 2.1　过渡方式编码表

编　　码	过 渡 方 式	编　　码	过 渡 方 式
0	盒状收缩	12	随意溶解
1	盒状放射	13	由左右往中间展开
2	图形收缩	14	由中间往左右展开
3	图形放射	15	由上下往中间展开
4	由下往上	16	由中间往左右展开
5	由上往下	17	由右下往左下展开
6	由左往右	18	由右下往左上展开
7	由右往左	19	由左上往右下展开
8	垂直百叶窗	20	由左下往右上展开
9	水平百叶窗	21	水平线状展开
10	水平格状百叶窗	22	垂直线状展开
11	垂直格状百叶窗	23	随机产生一种过渡方式

2.2.2　能力目标

能使用 meta 标记设置网页的编码方式、定时跳转以及过渡效果。

2.2.3　任务驱动

1. 任务的主要内容

编写两个网页：example2_2. html 和 example2_3. html。将这两个网页放在同一目录下，运行效果如图 2.1 和图 2.2 所示。具体要求如下。

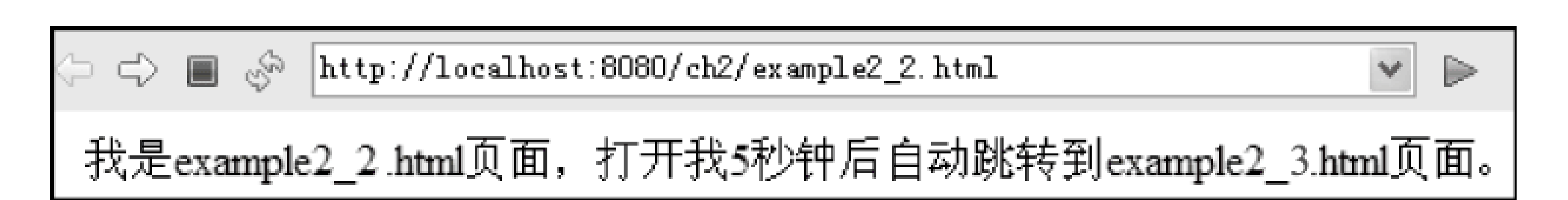


图 2.1　跳转前

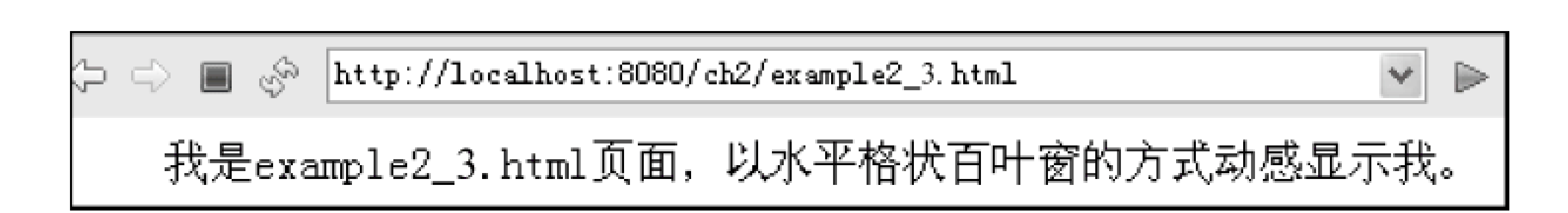


图 2.2　跳转到 example2_3. html 页面

- example2_2. html 的字符编码是 utf-8，example2_3. html 的字符编码是 gb2312。
- 打开 example2_2. html 页面 5 秒钟后自动跳转到 example2_3. html 页面。
- 从 example2_2. html 页面进入 example2_3. html 页面时，example2_3. html 页面以水平格状百叶窗的方式动感显示。

2. 任务的代码模板

将下列 example2_2. html 和 example2_3. html 中的【代码】替换为 HTML 代码。

example2_2. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;【代码 1】">
        <!--代码 1 设置编码为 utf-8 -->
        <meta http-equiv="【代码 2】" content="5;【代码 3】">
        <!--代码 2 与代码 3 设置定时跳转 -->
        <title>example2_2.html</title>
    </head>
    <body>
        <center>
            我是 example2_2.html 页面,打开我 5 秒钟后自动跳转到 example2_3.html 页面。
        </center>
    </body>
</html>
```

example2_3. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;【代码 4】">
        <!--代码 4 设置编码为 gb2312 -->
        <meta http-equiv="【代码 5】" content="revealtrans(duration=5,【代码 6】)"><!--代码
        5 设置过渡事件为进入页面,代码 6 设置过渡方式为 10-->
        <title>example2_3.html</title>
    </head>
    <body>
        <center>
            我是 example2_3.html 页面,以水平格状百叶窗的方式动感显示我。
        </center>
    </body>
</html>
```

3. 任务小结或知识扩展

使用 meta 标记实现从一个网页自动跳转到另一个网页时,要注意跳转网页文件的路径。如果只指定跳转网页的文件名,那么这两个网页文件必须在同一个目录中。

meta 标记很难记忆,不要担心,因为绝大多数人都记不住。即使是最老练和专业的编码人员也很难正确地记住这些标记,所以做得最多的一件事是,从正确的地方(最可能是一个项目中)直接将源代码复制过来。那些功能齐备的 Web 开发工具,如 Dreamweaver 通常会对这些难以记忆的标记自动生成代码。

在 head 标记中还会出现诸如 CSS 标记、JavaScript 代码之类的内容,不过这些东西将在需要时再做介绍。

4. 代码模板的参考答案

【代码 1】：charset=utf-8
【代码 2】：refresh
【代码 3】：url=example2_3.html
【代码 4】：charset=gb2312
【代码 5】：page-enter
【代码 6】：transition=10

2.2.4　实践环节

- 首先将任务中的【代码 1】换成"charset=ISO-8859-1"，然后运行 example2_2.html 网页，看看页面显示内容是否有乱码。
- 在 example2_2.html 网页的 head 标记中加上<meta http-equiv="page-enter" content=" revealtrans(duration=5, transition=0)">，再运行 example2_2.html 网页，看看运行效果。

2.3　使用 body 标记设置背景色和文字颜色

2.3.1　核心知识

HTML 页面的 body 标记中几乎包含了在浏览器中所看到的一切，如网页的背景、文字和图片等。这些内容都是通过 body 标记的属性进行设置的。

1. 设置网页的背景色

通常情况下，浏览器的背景色为白色，但使用 bgcolor 属性可以设置整个网页的背景色，设置背景色的格式如下：

```
<body bgcolor="网页的背景色">
```

其中，bgcolor 是一个属性，它的值可以是一个颜色英文名称，如 red，也可以是一个十六进制的颜色值，如 #FF0099。部分颜色编码值如表 2.2 所示。

表 2.2　部分颜色编码表

颜色	十六进制编码	英文名称	颜色	十六进制编码	英文名称
红	#FF0000	red	蓝	#0000FF	blue
橙	#FFA500	orange	靛	#4B0082	indigo
黄	#FFFF00	yellow	紫	#800080	purple
绿	#008000	green			

2. 设置网页的文字颜色

body 标记中的 text 属性可以设置网页文字的颜色，在没有对文字的颜色单独定义时，这一属性将对页面中所有的文字起作用，设置网页文字颜色的格式如下：

```
<body text="文字的颜色">
```

其中，text 的属性值与定义网页背景色相同。

2.3.2 能力目标

能使用 body 标记的 bgcolor 属性设置网页的背景色，能使用 body 标记的 text 属性设置网页的文字颜色。

2.3.3 任务驱动

1. 任务的主要内容

编写一个网页，该网页的背景色为蓝色，文字颜色为白色。运行效果如图 2.3 所示。

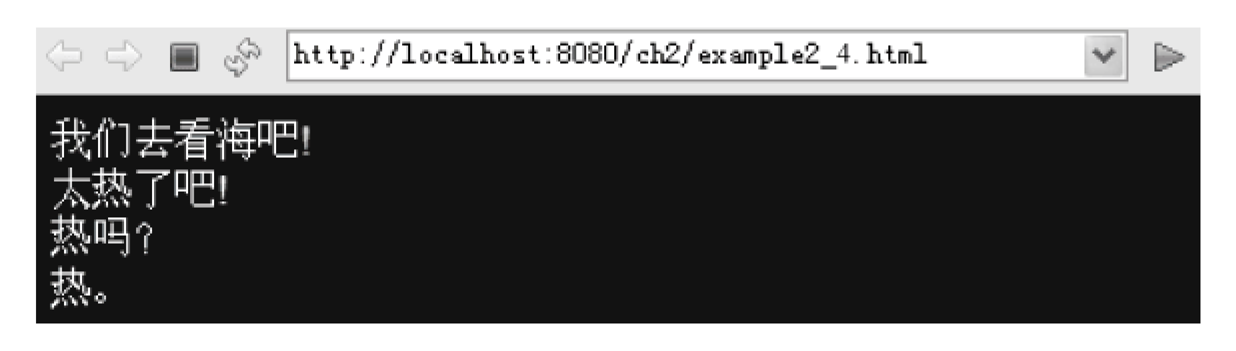


图 2.3　example2_4.html 页面的运行效果

2. 任务代码模板

将下列 example2_4.html 中的【代码】替换为 HTML 代码。

example2_4.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
        <title>example2_4.html</title>
    </head>
    <body【代码 1】【代码 2】>
        <!--代码 1 设置网页背景色为蓝色,代码 2 设置文字颜色为白色-->
        我们去看海吧!<br>太热了吧!<br>热吗? <br>热。
    </body>
</html>
```

3. 任务小结或知识扩展

在设置网页的背景色时，要尽量考虑到浏览者的感受。一般情况下，网页背景色都是浅颜色的，这样会减轻浏览者的视觉疲劳。另外，在设置网页中文字的颜色时，要注意文字颜色和网页背景色不要相同或相近。

4. 代码模板的参考答案

```
【代码 1】: bgcolor="blue"
【代码 2】: text="white"
```

2.3.4 实践环节

编写一个网页，背景颜色为绿色，网页中的文字颜色为白色，网页中的文字内容由读者

自行选择。

2.4　使用 body 标记设置背景图片和边距

2.4.1　核心知识

1. 设置网页的背景图片

除了使用 bgcolor 为网页定义背景色之外，还可以使用 background 属性为网页设置背景图片，格式如下：

```
<body background="背景图片的地址">
```

其中，background 的值是一个图片的地址。一般情况下，作为网页背景的图片会小于浏览器窗口的显示尺寸，此时浏览器就会横向或纵向重复显示该图片，直到铺满浏览器的窗口为止。如果在网页中既使用 bgcolor 设置了背景色，又使用 background 设置了背景图片，那么浏览器会用背景图片覆盖背景色。

2. 设置网页的边距

有时我们看到的网页内容并不是把整个浏览器占满，这是怎么制作的呢？在制作网页时，可以设置网页内容与浏览器左侧、上部边框之间的距离。格式如下：

```
< body topmargin="上边距的值" leftmargin="左边距的值" rightmargin="右边距的值"
bottommargin="下边距的值">
```

其中，边距的值就是普通的数字，默认以像素为单位。

2.4.2　能力目标

能使用 body 标记的 background 属性设置网页的背景图片，能使用 body 标记的 topmargin 属性和 leftmargin 属性设置网页的边距。

2.4.3　任务驱动

1. 任务的主要内容

编写一个网页，选择一张图片作为该网页的背景图片，背景图片与 HTML 网页文件在同一个目录下。设置该网页内容的上边距为 50，左边距为 50。网页设置边距前运行效果如图 2.4 所示，网页设置边距后运行效果如图 2.5 所示。



图 2.4　example2_5.html 页面边距设置前的运行效果

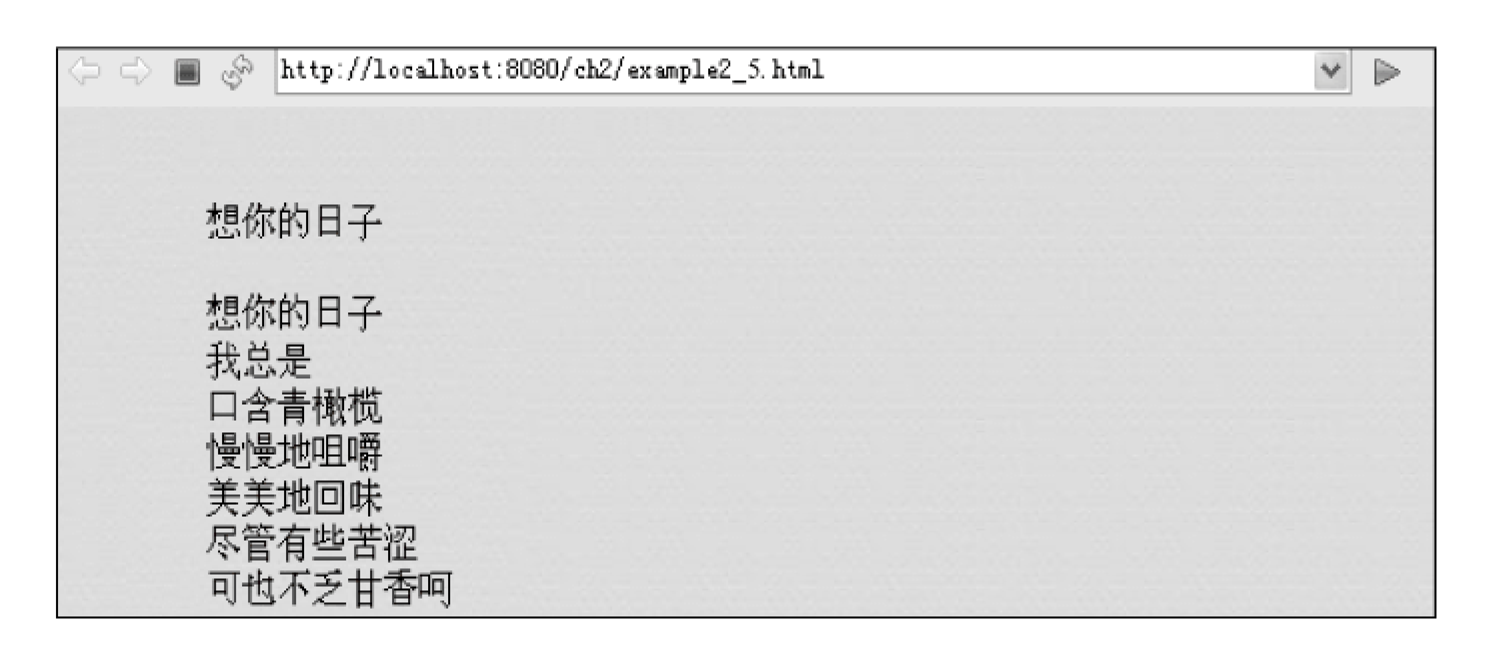


图 2.5　example2_5.html 页面边距设置后的运行效果

2. 任务代码模板

将下列 example2_5.html 中的【代码】替换为 HTML 代码。

example2_5.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
        <title>example2_5.html</title>
    </head>
        <body【代码 1】="beijingtupian.jpg"【代码 2】="50" leftmargin="50">
        <!--代码 1 设置网页背景图片,代码 2 设置网页上边距为 50-->
        想你的日子<br><br>想你的日子<br>我总是 <br>口含青橄榄 <br>慢慢地咀嚼 <br>
        美美地回味 <br>尽管有些苦涩 <br>可也不乏甘香呵
        </body>
</html>
```

3. 任务小结或知识扩展

从图 2.4 与图 2.5 看出,网页边距设置后网页背景的边距并没有改变。如果想让网页背景的边距也跟着变化的话,可以通过 CSS 来设置网页的页边距,CSS 知识将在后面章节讲解。

使用 body 标记的 background 属性设置网页背景图片时,要注意它的值是一个图片的地址。如果只指定图片文件的名字,那么要保证图片文件和网页文件在同一个目录中。

4. 代码模板的参考答案

【代码 1】: background
【代码 2】: topmargin

2.4.4 实践环节

编写一个网页,网页背景图片及文字内容由读者自行选择。网页的上边距的值设置为 100,左边距的值设置为 50,右边距的值设置为 100。

2.5 小　　结

在网页中，除了 head、title、meta 以及 body 这些基本标记外，还包括一种不显示在网页中的标记，那就是代码的注释。适当的注释可以更好地理解网页中的代码，也可以方便以后的维护，是一种良好的编程习惯。注释标记由"<!--"开始，以"-->"结束，开始部分是由一个左尖括号、一个半角感叹号和两个连字符组成，结束部分由两个连字符和一个右尖括号组成。

使用 meta 标记可以定义页面语言的编码方式，一般情况下，页面语言的编码方式都定义通用的编码 UTF-8。

在 body 标记中可以设置整个网页的背景色、背景图片以及文字颜色，但程序开发者要注意网页的整体美观，不要给浏览者造成视觉疲劳。

习　题　2

1. 以下标记中，用于设置页面标题的是(　　)。

   A. <title>　　B. <caption>　　C. <head>　　D. <html>

2. 若是设计网页的背景图形为 bg. png，以下标记中，正确的是(　　)。

   A. <body background="bg. png">

   B. <body bground="bg. png">

   C. <body image="bg. png">

   D. <body bgcolor="bg. png">

3. 关于 HTML 文件的说法正确的是(　　)。

   A. HTML 标记都必须配对使用

   B. 在<title>和</title>标签之间的是头信息

   C. HTML 标签是区分大小写的，<B>跟<b>表示的意思是不一样的

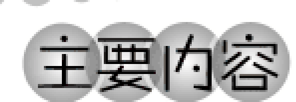
   D. <!-- -->标记是注释标记

4. 编写一个网页，具体要求如下：
   - 标题为"我自己的空间"；
   - 文字编码为 gb2312；
   - 以垂直格状百叶窗的方式动感显示；
   - 背景色为靛色，网页中的文字颜色为红色；
   - 上边距的值设置为 200，左边距的值设置为 100，右边距的值设置为 100；
   - 文字内容由读者自行选择。

5. 编写两个网页：index. html 和 autopage. html，具体要求如下。
   - index. html 网页运行 10 秒钟后，自动打开 autopage. html 网页；
   - 由读者自行选择一张图片作为 index. html 网页的背景图片；
   - 两个网页中的文字内容由读者自行选择。



第 3 章 文字布局与文字效果



主要内容

- 标题标记
- 文字格式
- 段落格式
- 文字滚动
- 水平线

在一个网页中,文字是灵魂,它可以起到传递信息、导航以及交互的作用。在网页中显示大量的文字并不困难,但是如何对这些文字进行编排以及控制这些文字的显示方式,让网页浏览者舒适地浏览网页,是一个比较头疼的问题。本章根据这个问题,主要讲述文字格式、段落格式、文字滚动以及水平线的设计。

3.1 标题标记

3.1.1 核心知识

HTML 将和文本相关的标题分成 6 个级别,1 至 6 级的标题语法格式如下:

```
<h1> ...</h1>
<h2> ...</h2>
<h3> ...</h3>
<h4> ...</h4>
<h5> ...</h5>
<h6> ...</h6>
```

h1 到 h6,作为标题标记,依据重要性递减。遵循这样的原则是很有必要的,它能让页面的层级关系更清楚,让搜索引擎更好地抓取和分析出页面的主题内容等。浏览器的默认样式也是依据标题的重要性递减,字号从 h1 到 h6 由大变小。为了更好地理解,请看下面的代码段:

```
<h1>学习标题标记</h1>
    <h2>第 3 章文本内容</h2>
        <h3>3.1 文字</h3>
            <h4>3.1.1 字号</h4>
```

h1 一级标题代表重中之重，一般运用于网站标题或者头条新闻。虽然 h1 代码可以写多个，但实际上有唯一性，h1 标题在一个页面里面最好只出现 1 次或者没有。h2 二级标题主要出现在页面主体内容的文章标题和栏目标题上。h3 三级标题一般出现在页面的边侧栏上。页面层级标题不能太多，所以 h4、h5 和 h6 一般出现得较少。

3.1.2　能力目标

能使用标题标记设置 HTML 页面各种级别的标题。

3.1.3　任务驱动

1. 任务的主要内容

编写一个网页，网页中包含 1、3、5 三个级别标题。网页运行效果如图 3.1 所示。

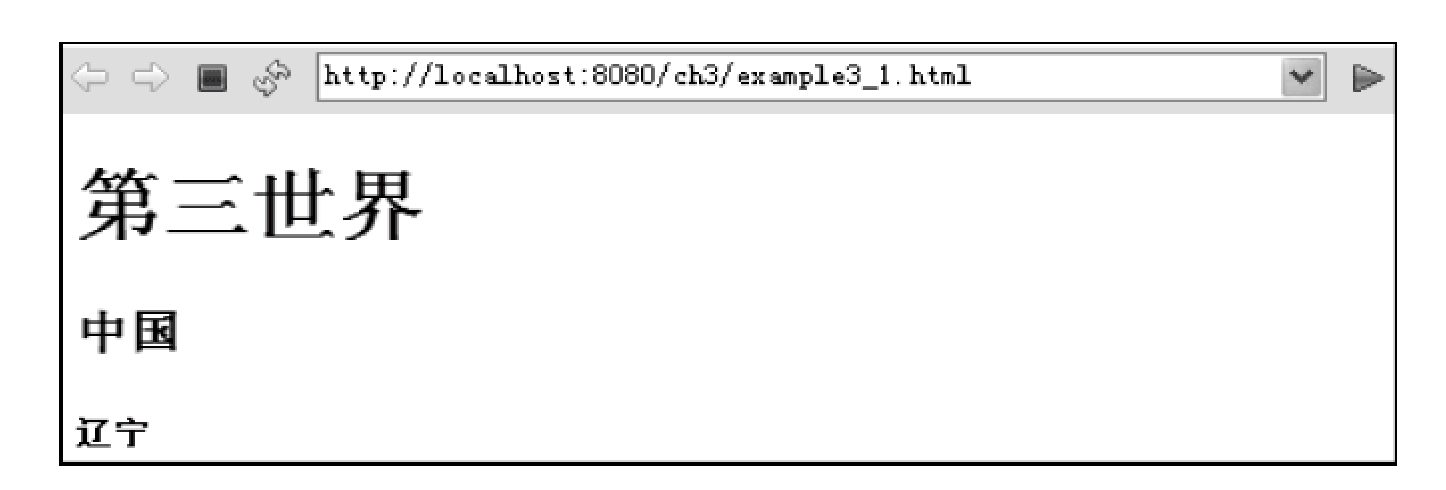


图 3.1　example3_1.html 运行效果

2. 任务的代码模板

将下列 example3_1.html 中的【代码】替换为 HTML 代码。

example3_1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_1.html</title>
</head>
<body>
    【代码 1】<!--设置 h1 标题为“第三世界”-->
    【代码 2】<!--设置 h3 标题为“中国”-->
    【代码 3】<!--设置 h5 标题为“辽宁”-->
</body>
</html>
```

3. 任务小结或知识扩展

不同的浏览器，标题的字体大小是不相同的。这种差别，通常在布局中用 CSS 将其统一。

4. 代码模板的参考答案

【代码 1】：<h1>第三世界</h1>

【代码 2】：<h3>中国</h3>
【代码 3】：<h5>辽宁</h5>

3.1.4 实践环节

编写一个网页，网页中包含 2、3、4 三级标题，网页运行效果如图 3.2 所示。

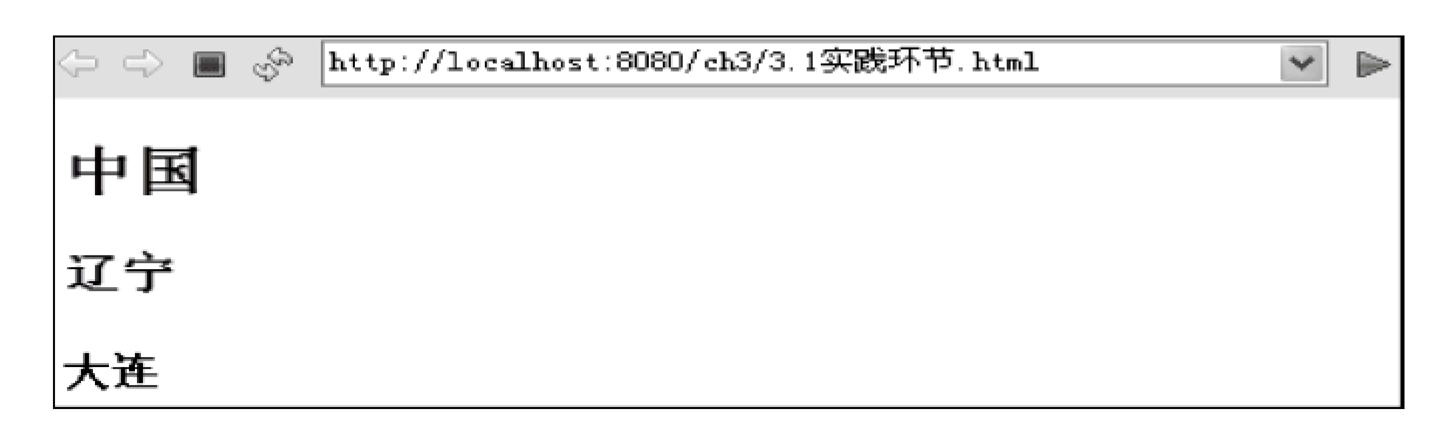


图 3.2　3.1 实践环节.html 运行效果

3.2　使用 font 标记设计文字的大小、字体以及颜色

3.2.1 核心知识

1. 设置文字大小

文字的大小在网页中非常重要，它是引起浏览者注意的一个因素，使用<font>标记中的 size 属性可以设置文字的大小，语法格式如下：

```
<font size="文字大小">被设置的文字</font>
```

其中，size 属性的值有相对与绝对两种定义，从 1 到 7 的整数，代表字体大小的绝对字号；而从−4 到+4 的整数(不包含 0)的字体是相对于 3 号的放大与缩小字号，其中“+”号不能省略。

2. 设置文字字体

font 标记中的属性 face 用于设置网页中文字的字体名称，浏览器中只有安装了设置的字体，才能正确显示设置的字体。设置字体语法格式如下：

```
<font face="字体的名称">被设置的文字</font>
```

其中，face 属性值可以有 1 个或多个。默认情况下，使用第一种字体显示，如果浏览器没有第一种字体，则使用第 2 种，以此类推。如果设置的字体都不存在，则会以默认的字体显示。

3. 设置文字颜色

font 标记中的属性 color 用于设置网页中文字的颜色，语法格式如下：

```
<font color="字体的颜色">被设置的文字</font>
```

其中，字体的颜色值与网页背景色值一样，可以是一个英文单词，也可以是一个十六进制的颜色值。

3.2.2 能力目标

能使用 font 标记的属性 size 设置文字的大小，能使用 font 标记的属性 face 设置文字的字体类型，能使用 font 标记的属性 color 设置文字的颜色。

3.2.3 任务驱动

1. 任务的主要内容

编写一个网页，网页中的文字大小为“6”号，字体类型为“华文行楷”，颜色为“blue”。网页运行效果如图 3.3 所示。



图 3.3　example3_2. html 运行效果

2. 任务的代码模板

将下列 example3_2. html 中的【代码】替换为 HTML 代码，有关代码要求：【代码 1】设置文字大小为 6 号，【代码 2】设置文字字体类型为“华文行楷”，【代码 3】设置文字颜色为蓝色。

example3_2. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_2.html</title>
</head>
<body>
    <center>
        <font【代码 1】="6"【代码 2】="华文行楷"【代码 3】="blue">
            面朝大海，春暖花开。
        </font>
    </center>
</body>
</html>
```

3. 任务小结或知识扩展

如果网页中的文字大小、字体颜色都一样，那么整个版面看起来就会显得很单调。在网页中文字大小、字体以及颜色设置是相当重要的，只有大小、字体以及颜色丰富而协调才能吸引浏览者的注意，并得到认可。但要注意，网页的文字格式不要过于丰富，不要给浏览者带来视觉疲劳。

4. 代码模板的参考答案

【代码 1】：size

【代码 2】：face
【代码 3】：color

3.2.4 实践环节

编写一个网页，网页中的文字大小为"＋1"号，字体类型为"黑体"，颜色为"green"。网页运行效果如图 3.4 所示。

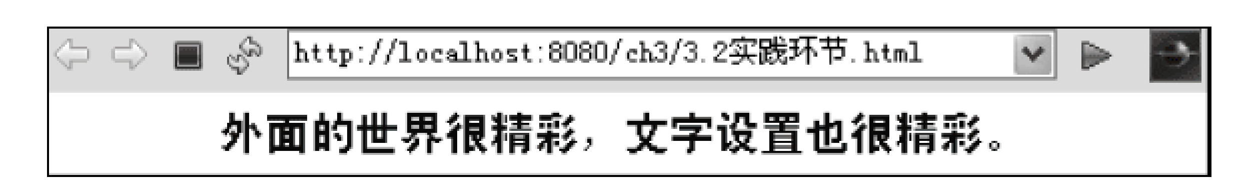


图 3.4　3.2 实践环节网页运行效果

3.3　粗体与斜体

3.3.1 核心知识

1. 设置文字粗体

和是网页中设置粗体文字的基本元素。在和或在和之间的文字，在浏览器中都会以粗体字体显示，语法格式如下：

```
<b>被加粗的文字</b>
<strong>被加粗的文字</strong>
```

这两个元素的首尾标记都是必需的，如果没有结尾，浏览器会认为从标记开始的所有文字都是粗体。

2. 设置文字斜体

<i>、和<cite>是网页中设置斜体文字的基本元素。在<i>和</i>、和或<cite>和</cite>之间的文字，在浏览器中都会以斜体字体显示，语法格式如下：

```
<i>斜体的文字</i>
<em>斜体的文字</em>
<cite>斜体的文字</cite>
```

这三个元素的首尾标记都是必需的，如果没有结尾，浏览器会认为从标记开始的所有文字都是斜体。

3.3.2 能力目标

能使用粗体标记设置粗体的文字，能使用斜体标记设置斜体的文字。

3.3.3 任务驱动

1. 任务的主要内容

编写一个网页，网页中的文字字体既有粗体，又有斜体。网页运行效果如图 3.5 所示。

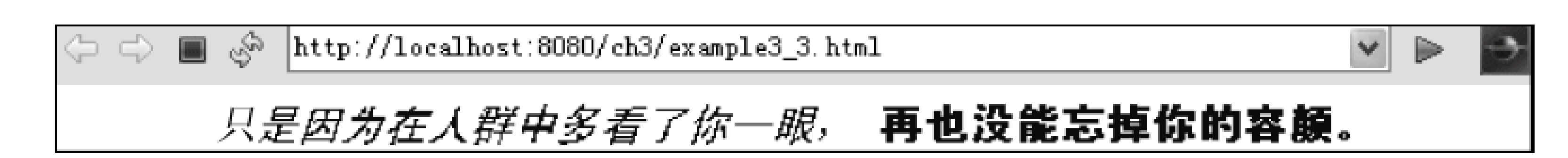


图 3.5　example3_3.html 运行效果

2. 任务的代码模板

将下列 example3_3.html 中的【代码】替换为 HTML 代码。

example3_3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_3.html</title>
</head>
<body>
    <center>
        <font size="4"【代码 1】="黑体" color="blue"><!--设置文字字体类型为"黑体"-->
        <i>只是因为在人群中多看了你一眼,【代码 2】<!--使用<i>标记设置斜体文字-->
        【代码 3】再也没能忘掉你的容颜。</strong><!--使用<strong>标记设置粗体文字-->
        </font>
    </center>
</body>
</html>
```

3. 任务小结或知识扩展

网页中的粗体与斜体文字起到醒目、强调或区别的作用。一段文字既可以设置斜体，同时又可以设置粗体，如<i>既斜又粗的文字</i>。

4. 代码模板的参考答案

```
【代码 1】: face
【代码 2】: </i>
【代码 3】: <strong>
```

3.3.4 实践环节

编写一个网页，网页中的文字大小为“5”号，字体为“华文宋体”，颜色为“red”，部分文字加粗，部分文字为斜体。网页运行效果如图 3.6 所示。

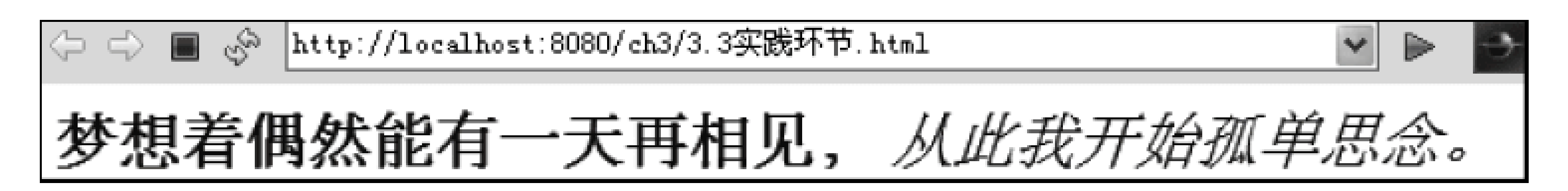


图 3.6　3.3 实践环节网页运行效果

3.4　上标与下标

3.4.1　核心知识

在一个网页中可能有数学公式、日常计算、书籍文章注解等内容，这时，就需要使用上标与下标将文字以小字体方式显示在另一段文字的右上角或右下角。上标与下标语法格式如下：

```
<sup>上标内容</sup>
<sub>下标内容</sub>
```

这两种标记的用法基本相同，只需将文字放在标记中间即可。

3.4.2　能力目标

能使用上标标记<sup>设置文字的上标，能使用下标标记<sub>设置文字的下标。

3.4.3　任务驱动

1. 任务的主要内容

编写一个网页，网页内容有个一元三次方程 $ax^3+bx^2+cx+d=0$。网页运行效果如图3.7所示。

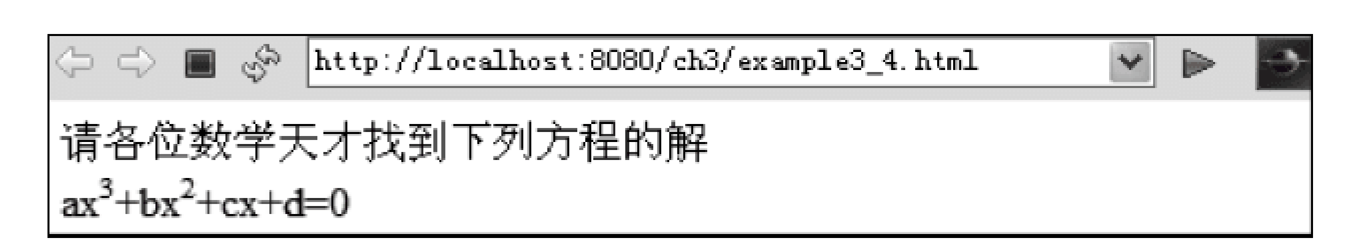


图3.7　example3_4.html运行效果

2. 任务的代码模板

将下列example3_4.html中的【代码】替换为HTML代码。

example3_4.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_4.html</title>
</head>
    <body>
        请各位数学天才找到下列方程的解<br>
        ax【代码 1】+bx【代码 2】+cx+d=0 <!--代码 1 设置上标为 3,代码 2 设置上标为 2-->
    </body>
</html>
```

3. 任务小结或知识扩展

上标与下标的用法非常简单，只要认识什么是上标，什么是下标，就可以运用上标标记与下标标记制作出所需要的网页。

4. 代码模板的参考答案

【代码 1】: ³
【代码 2】: ²

3.4.4 实践环节

编写一个网页，网页内容有个一元二次方程 $x^2+x-2=0$ 以及它的解。网页运行效果如图 3.8 所示。

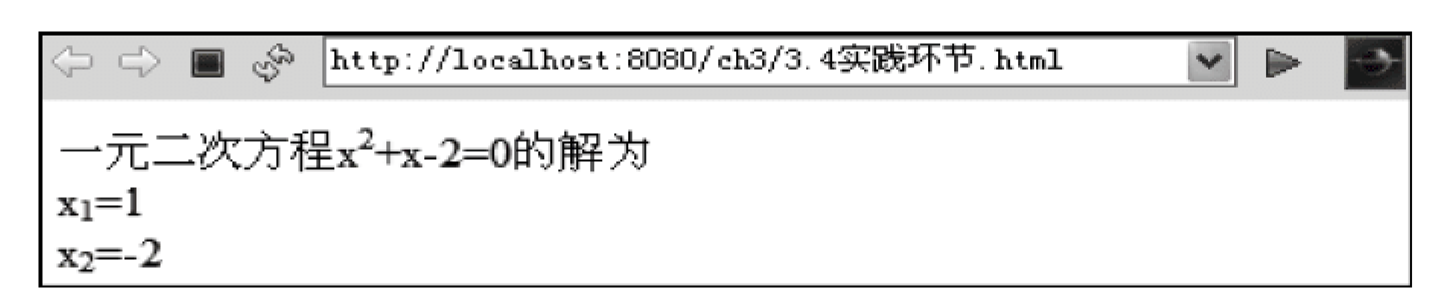


图 3.8　3.4 实践环节网页运行效果

3.5　删除线与下画线

3.5.1 核心知识

1. 设置文字删除线

在网页中有时候需要给文字添加删除线以便突出显示，<strike>和<s>标记都可以给文字添加删除线，语法格式如下：

```
<strike>删除的文字</strike>
<s>删除的文字</s>
```

2. 设置文字下画线

在网页中有时候需要给文字添加下画线以表示该文字重要或突出显示。<u>和<ins>标记都可以给文字添加下画线，语法格式如下：

```
<u>有下画线的文字</u>
<ins>有下画线的文字</ins>
```

3.5.2 能力目标

能使用<u>或<ins>标记给文字添加下画线，能使用<strike>或<s>标记给文字添加删除线。

3.5.3 任务驱动

1. 任务的主要内容

编写一个网页，在网页中使用下画线标注出自己所居住过的城市，用删除线标注出不是自己所居住过的城市，网页运行效果如图3.9所示。

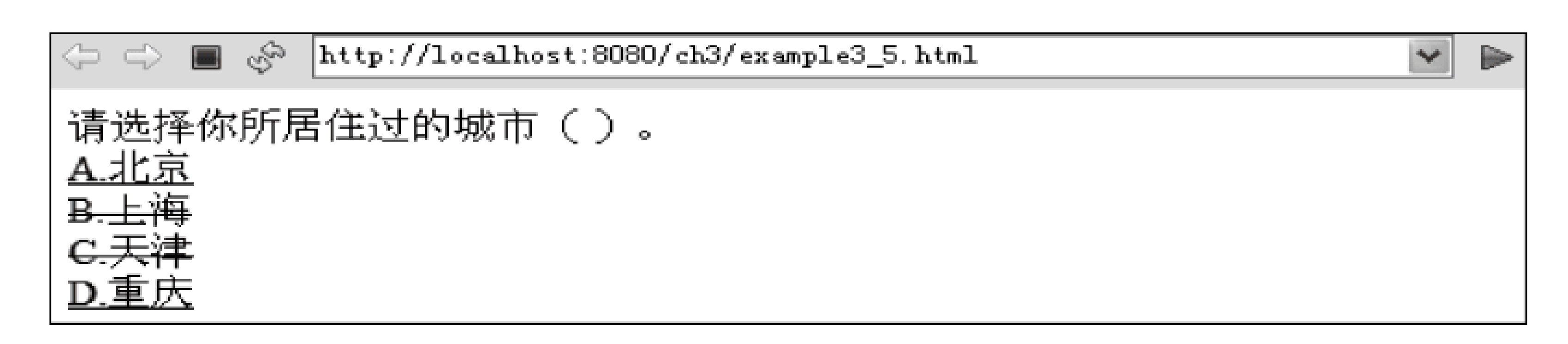


图3.9　example3_5.html运行效果

2. 任务的代码模板

将下列example3_5.html中的【代码】替换为HTML代码。

example3_5.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_5.html</title>
</head>
<body>
    请选择你所居住过的城市(   )。<br>
    【代码 1】</u><br>          <!--使用标记 u 给选项“A.北京”添加下画线-->
    <s>B.上海</s><br>
    <strike>【代码 2】<br>    <!--使用标记 strike 给选项“C.天津”添加删除线-->
    <ins>D.重庆</ins>
</body>
</html>
```

3. 任务小结或知识扩展

从任务中可以看出<s>和<strike>标记是一样的，都是添加删除线；<u>和<ins>标记是一样的，都是添加下画线。删除线标记与下画线标记的用法相似，只需把要设置的文字放在标记中间即可。在网页制作时，使用删除线标记和下画线标记的主要目的是使文字突出显示。

4. 代码模板的参考答案

【代码1】：<u>A.北京
【代码2】：C.天津</strike>

3.5.4 实践环节

编写一个网页，在网页中使用删除线标出计算错误的算式，使用下画线标出计算正确的

算式。网页运行效果如图 3.10 所示。

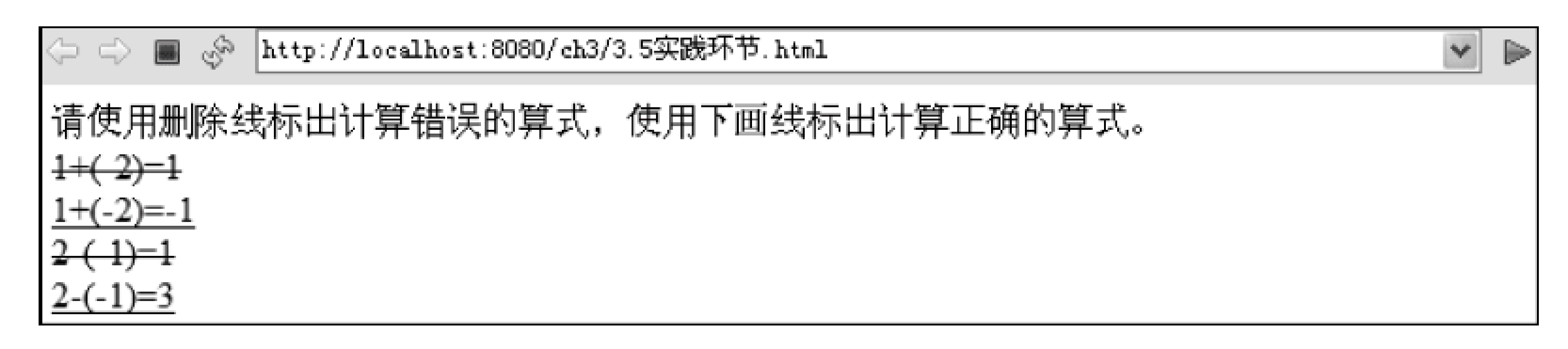


图 3.10　3.5 实践环节网页运行效果

3.6　使用 p 标记设计段落格式

3.6.1　核心知识

1. 使用<p> 开始一个新段落

制作网页时，经常需要开始一个新段落编写文本。在 HTML 网页中，使用<p>标记实现一个新段落，语法格式如下：

```
<p>段落的内容</p>
```

开始一个新段落可以使用成对的<p>标记包含一个段落，也可以使用单独的<p>标记来划分段落。

2. 设置一个段落文字的对齐方式

<p>标记中的属性 align 能够设置段落中文字的对齐方式，对齐方式分为：左对齐、居中和两端对齐，语法格式如下：

```
<p align="对齐方式"></p>
```

其中，align 的取值为 left 时，文字显示左对齐；align 的取值为 right 时，文字显示右对齐；align 的取值为 center 时，文字显示居中对齐。

3. 设置段落的文字方向

在制作网页时，有些特殊的文字需要特殊的书写方向。<bdo>标记就可以实现文字书写方向的改变，语法格式如下：

```
<bdo dir="文字方向">文字内容</bdo>
```

其中，文字方向是由 dir 属性指定的，dir 属性值有 ltr(从左到右)和 rtl(从右到左)两种，默认文字方向为从左到右。

4. 空字符

在 HTML 网页中使用空格键是无效的(因为多少个空格键浏览器都看成一个空格)，这时需要使用空字符“ ”。空字符“ ”里面的符号和字母都是半角状态的。

5. 换行标记

换行标记
在不另起一段的情况下将当前文本强制换行。该标记是一个没有结尾

的标记，HTML 网页中任何位置只要使用了
标记，该位置之后的文字在浏览器中都会在下一行显示。

3.6.2 能力目标

能使用<p>标记设置段落以及设置段落文字的对齐方式，能使用<bdo>标记设置段落的文字方向，能使用空字符“&nsp;”设置空格，能使用换行标记
强制换行。

3.6.3 任务驱动

1. 任务的主要内容

编写一个网页，网页中一共有三段文字。第一段文字对齐方式为居左，文字方向为从左到右；第二段文字对齐方式为居中，文字方向为从左到右；第三段文字对齐方式为居右，文字方向为从右到左。网页运行效果如图 3.11 所示。

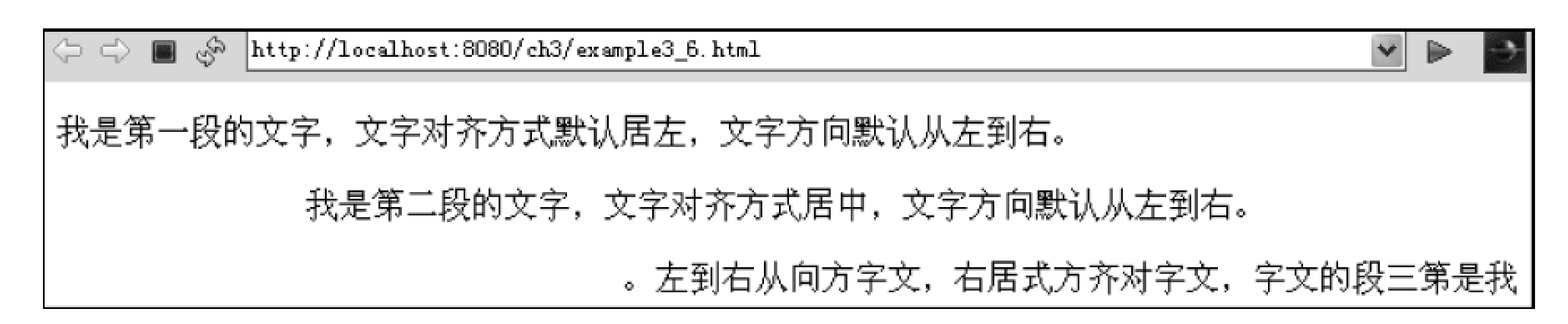


图 3.11　example3_6. html 运行效果

2. 任务的代码模板

将下列 example3_6. html 中的【代码】替换为 HTML 代码。

example3_6. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_6.html</title>
</head>
<body>
    <p>我是第一段的文字,文字对齐方式默认居左,文字方向默认从左到右。</p>
    <!--代码 1 使用 align 设置文字居中对齐 -->
    <p【代码 1】>我是第二段的文字,文字对齐方式居中,文字方向默认从左到右。</p>
    <!--代码 2 使用 align 设置文字居右对齐,代码 3 使用 dir 设置文字方向为从右到左-->
    <p【代码 2】><bdo【代码 3】>我是第三段的文字,文字对齐方式居右,文字方向从右到左。
      </bdo></p>
</body>
</html>
```

3. 任务小结或知识扩展

从任务中可以看出段落中文字的对齐方式默认为居左，文字方向默认为从左到右。其实，大多数情况下段落中的文字都是采用默认的格式，偶尔需要指定特殊的文字才进行段落文字的格式设置。

4. 代码模板的参考答案

【代码 1】: align="center"
【代码 2】: align="right"
【代码 3】: dir="rtl"

3.6.4　实践环节

编写一个网页，网页中有两段文字，一段为古诗的标题和作者，一段为古诗的内容，两段文字对齐方式都为居中。网页运行效果如图 3.12 所示。

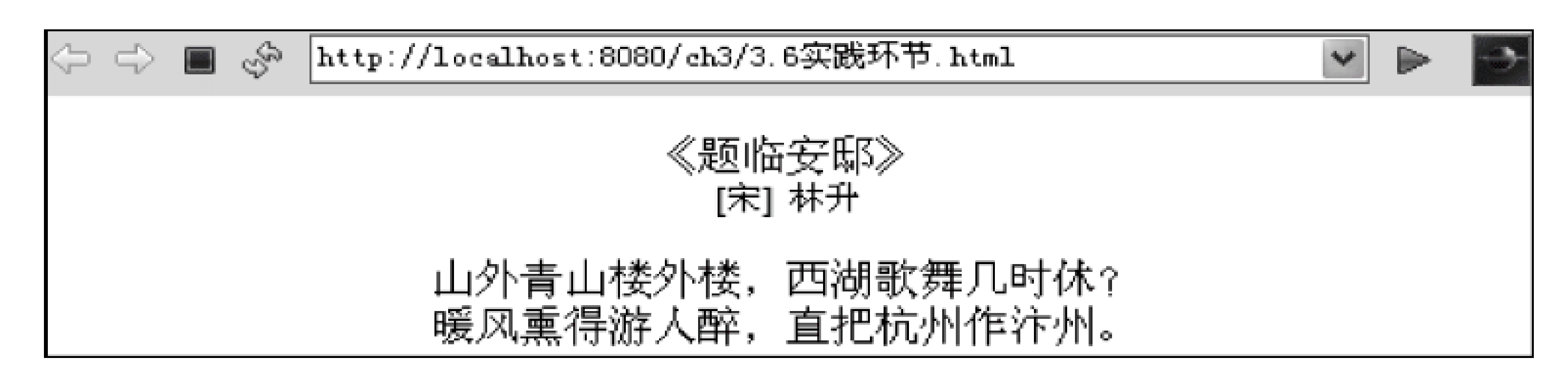


图 3.12　3.6 实践环节网页运行效果

3.7　水　平　线

3.7.1　核心知识

水平线在网页中可以分隔网页的内容，使网页内容结构清晰、层次分明，便于浏览。使用<hr>就可以在网页中添加一条水平线，来分隔不同的文字内容，语法格式如下：

```
<hr width="宽度" size="高度" color="水平线的颜色" align="对齐方式" noshade>
```

其中，width 属性值可以是绝对的像素值，也可以是相对的百分比；size 属性值只能使用绝对的像素值来定义；color 属性用来设置水平线的颜色；align 属性用来设置水平线的对齐方式；noshade 可以在网页中去掉水平线的阴影。

3.7.2　能力目标

能使用<hr>标记及它的属性来设置所需要的水平线。

3.7.3　任务驱动

1. 任务的主要内容

编写一个网页，网页中一共有三段文字。第一段文字与第二段文字之间有条水平线，该水平线的宽度为 500 像素，高度为 2 像素，颜色为红色，对齐方式居中。第二段文字与第三段文字之间有条水平线，该水平线的宽度为 800 像素，高度为 5 像素，颜色为绿色，对齐方式居中。网页运行效果如图 3.13 所示。

2. 任务的代码模板

将下列 example3_7.html 中的【代码】替换为 HTML 代码。

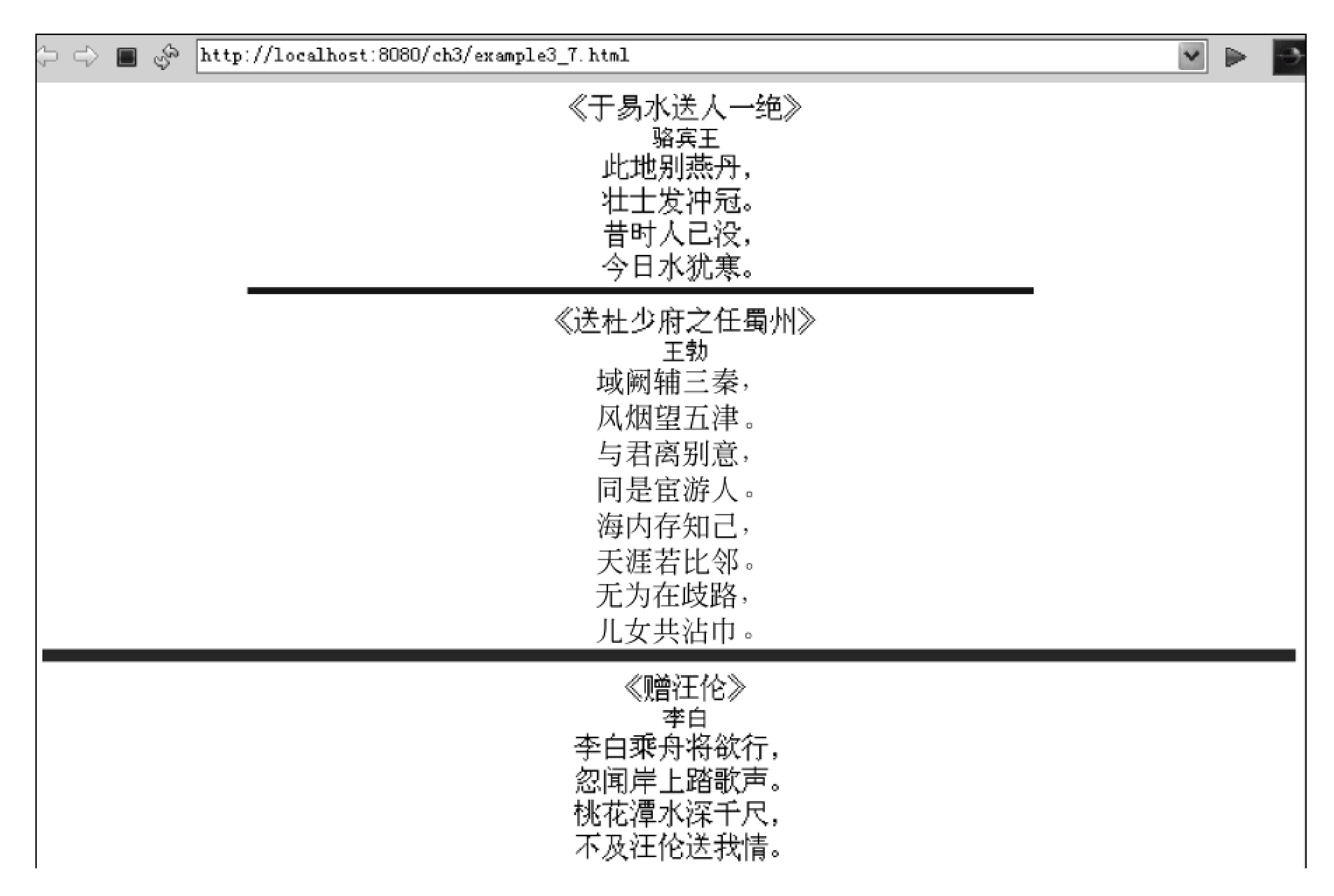


图 3.13　example3_7.html 运行效果

example3_7.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_7.html</title>
</head>
<body>
    <p align="center">《于易水送人一绝》<br><font size="2">骆宾王</font><br>
        此地别燕丹,<br>
        壮士发冲冠。<br>
        昔时人已没,<br>
        今日水犹寒。<br>
    </p>
    【代码 1】<!--使用 hr 标记设置一条宽度为 500 像素,高度为 2 像素,颜色为红色,对齐方式居中
                的水平线 -->
    <p align="center">《送杜少府之任蜀州》<br><font size="2">王勃</font><br>
        城阙辅三秦,<br>
        风烟望五津。<br>
        与君离别意,<br>
        同是宦游人。<br>
        海内存知己,<br>
        天涯若比邻。<br>
        无为在歧路,<br>
        儿女共沾巾。<br>
    </p>
    【代码 2】<!--使用 hr 标记设置一条宽度为 800 像素,高度为 5 像素,颜色为绿色,对齐方式居中
```

```
            的水平线 -->
    <p align="center">《赠汪伦》<br><font size="2">李白</font><br>
        李白乘舟将欲行,<br>
        忽闻岸上踏歌声。<br>
        桃花潭水深千尺,<br>
        不及汪伦送我情。<br>
    </p>
</body>
</html>
```

3. 任务小结或知识扩展

从任务中可以看出在段落文字之间使用水平线可以使文字内容结构清晰、层次分明。在没有指定水平线的对齐方式时,默认为居中。在没有指定水平线的颜色时,默认为灰黑色。

4. 代码模板的参考答案

【代码 1】: `<hr width="500" size="2" color="red" align="center">`
【代码 2】: `<hr width="800" size="5" color="green" align="center">`

3.7.4　实践环节

编写一个网页,网页中有两段文字,两段文字之间有条水平线,该水平线的宽度为 700 像素,高度为 4 像素,颜色为蓝色,对齐方式默认,去掉水平线的阴影。网页运行效果如图 3.14 所示。

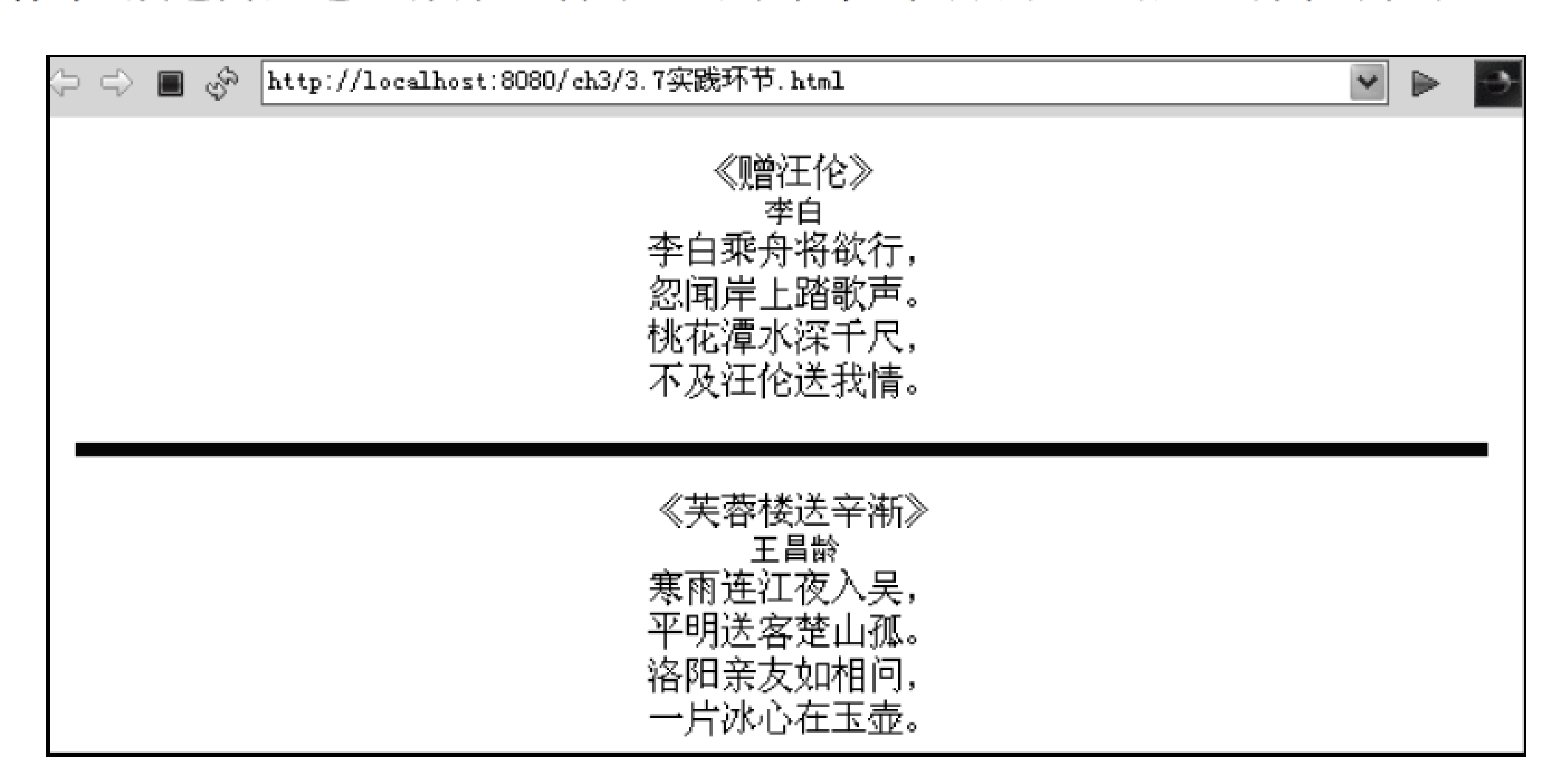


图 3.14　3.7 实践环节网页运行效果

3.8　文 字 滚 动

3.8.1　核心知识

1. 滚动方向

文字滚动的方向,默认情况下是从右向左的,但可以通过<marquee>标记里的 direction 属性来改变文字滚动的方向,语法格式如下:

```
<marquee direction="滚动方向">滚动的文字</marquee>
```

其中，direction 的值有 up、down、left 和 right 四个值，分别表示向上、向下、向左和向右滚动，向左滚动就是默认情况。

2. 滚动方式

文字滚动的方式，默认情况下是循环滚动，但可以通过<marquee>标记里的 behavior 属性来改变文字滚动的方式，语法格式如下：

```
<marquee behavior="滚动方式">滚动的文字</marquee>
```

其中，behavior 的值有 scroll、slide 和 alternate 三个，分别表示循环滚动、只滚动一次和来回交替滚动。

3. 滚动次数

文字滚动的方式，默认情况下是循环滚动的，如果希望滚动几次就停止，可以通过<marquee>标记里的 loop 属性来设置滚动的次数，语法格式如下：

```
<marquee loop="滚动次数">滚动的文字</marquee>
```

其中，loop 的值为整数。

4. 滚动速度

<marquee>标记里的 scrollamount 属性可以设置文字滚动的快慢，语法格式如下：

```
<marquee scrollamount="滚动速度">滚动的文字</marquee>
```

其中，滚动速度实际上是设置文字每次移动的长度，以像素为单位。

5. 滚动延迟

<marquee>标记里的 scrolldelay 属性可以设置文字滚动的时间间隔，语法格式如下：

```
<marquee scrolldelay="时间间隔">滚动的文字</marquee>
```

其中，滚动时间间隔单位是毫秒，如果设置的时间比较长，会产生走走停停的效果。

6. 滚动区域的背景颜色

<marquee>标记里的 bgcolor 属性可以设置文字滚动区域的背景色，以显示突出，语法格式如下：

```
<marquee bgcolor="背景颜色">滚动的文字</marquee>
```

其中，bgcolor 的值可以是一个已命名的颜色，也可以是一个十六进制的颜色值。

7. 滚动背景宽度和高度

默认情况下，滚动文字的背景与文字同高，与浏览器窗口同宽，使用 width 和 height 标记可以设置水平和垂直的范围，语法格式如下：

```
<marquee width="背景宽度" height="背景高度">滚动的文字</marquee>
```

其中，背景宽度和背景高度都是以像素为单位的。

3.8.2 能力目标

能使用<marquee>标记里的各种属性设置文字滚动效果。

3.8.3 任务驱动

1. 任务的主要内容

编写一个网页，网页中有一段滚动的文字，文字滚动方向为默认方向，文字滚动的背景色为蓝色，文字颜色为白色，文字滚动方式为来回交替滚动。网页运行效果如图 3.15 所示。

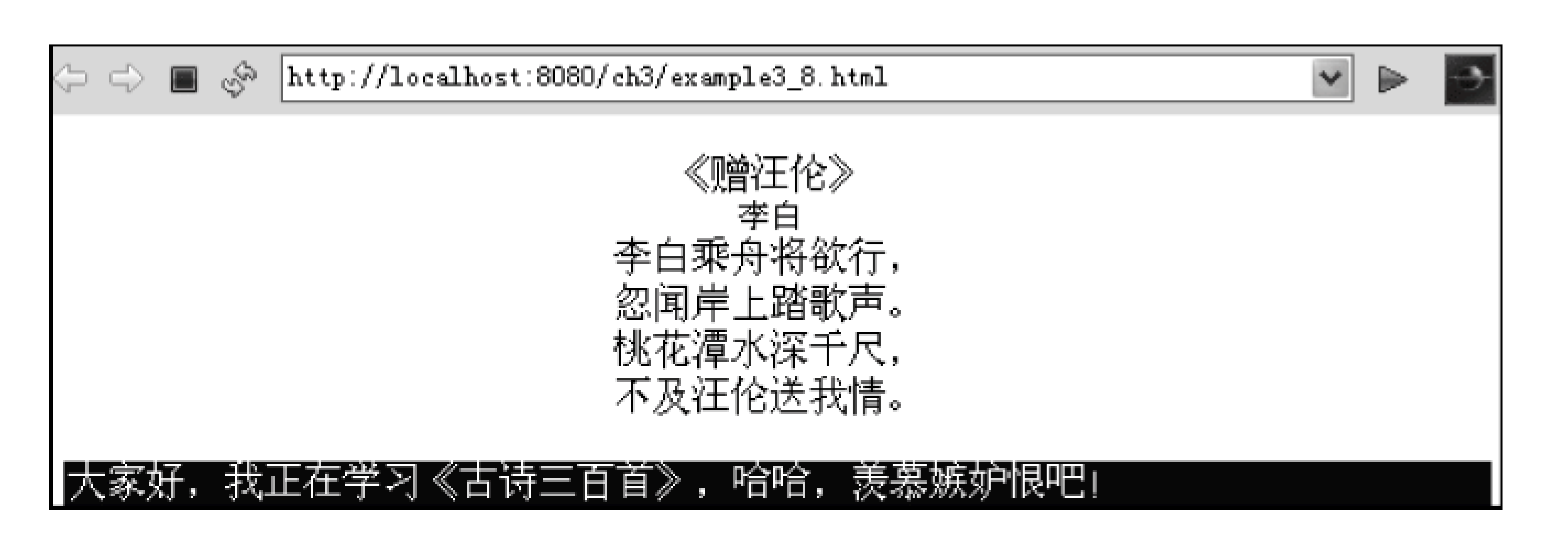


图 3.15　example3_8.html 运行效果

2. 任务的代码模板

将下列 example3_8.html 中的【代码】替换为 HTML 代码。

example3_8.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example3_8.html</title>
</head>
<body>
    <p align="center">《赠汪伦》<br><font size="2">李白</font><br>
        李白乘舟将欲行，<br>
        忽闻岸上踏歌声。<br>
        桃花潭水深千尺，<br>
        不及汪伦送我情。<br>
    </p>
    <marquee bgcolor="blue"【代码 1】><!--代码 1 使用 behavior 设置文字滚动方式为来回交替
                                                    -->
        <!--代码 2 设置文字颜色为白色-->
        <font【代码 2】>大家好，我正在学习《古诗三百首》，哈哈，羡慕嫉妒恨吧！</font>
    </marquee>
</body>
</html>
```

3. 任务小结或知识扩展

目前学习的文字滚动效果，只是简单的滚动。要想做出非常漂亮的滚动效果，等学习完网页布局、CSS 和 JavaScript 就可以。

4. 代码模板的参考答案

【代码 1】: behavior="alternate"
【代码 2】: color="white"

3.8.4 实践环节

编写一个滚动的公告板,具体要求如下:

- 公告板的内容字体类型是楷体_GB2312;
- 公告板的内容要不停地从下向上来回交替滚动,网页运行效果如图 3.16 所示。

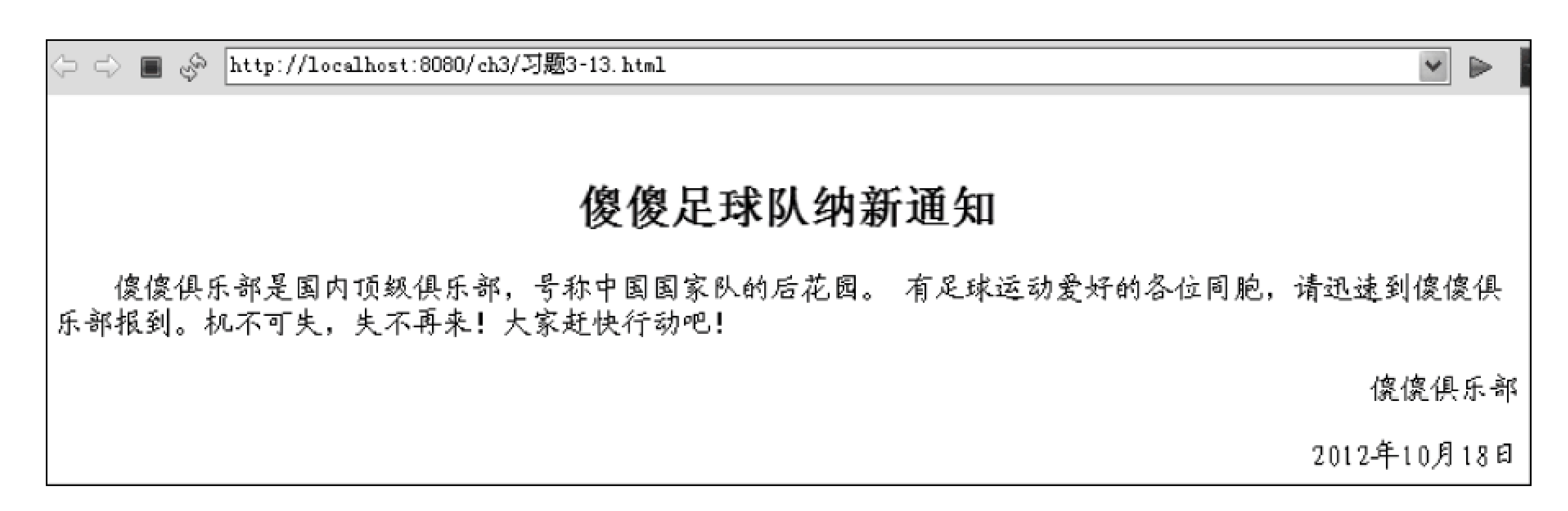


图 3.16　3.8 实践环节网页运行效果

3.9 小　　结

设计网页中文本内容时所使用的标记除了本章介绍的<h>、、<p>、<strong>、<strike>、<hr>、<marquee>等常用标记外,还有个别不常用的标记,如<nobr>、<pre>等。这些没有介绍的标记,读者可以查阅相关资料自主学习。

本章的重点是如何使用<h>、、<p>、<hr>、<marquee>等标记设计文本内容。

习　题　3

1. 下面哪一个属性不是文本的标记属性?(　　)
   A. nbsp　　B. align　　C. color　　D. face
2. 下面哪一项是换行标记?(　　)
   A. <body>　　B. 　　C.
　　D. <p>
3. 在 HTML 中,标记的 size 属性最大取值可以是(　　)。
   A. 5　　B. 6　　C. 7　　D. 8
4. 以下标记中,没有对应的结束标记的是(　　)。
   A. <body>　　B.
　　C. <html>　　D. <title>
5. 若要以加粗宋体、12 号字显示"vbscript",以下选项中正确的是(　　)。
   A. vbscript

B. vbscript

C. vbscript

D. vbscript

6. 以下标记中，可用来产生滚动文字的是(　　)。

A. <Scroll>　　B. <Marquee>

C. <TextArea>　　D. <IFRAME>

7. 下列关于 HTML 标题说法错误的是(　　)。

A. 标题标记为<hn>，其中 n 为标题的等级

B. HTML 提供 6 个等级的标题，n 越小，标题的字号就越大

C. HTML 必须出现标题标记

D. HTML 可以不出现标题标记

8. 下面说法中错误的是(　　)。

A. <p></p>标记对用来创建一个段落

B.
标记用来创建一个回车换行

C. <p>可以有多种属性

D. 换行标记
也有结束标记</br>

9. 下面说法中正确的是(　　)。

A. 我们只能通过设置绝对字号的方法，设置字体大小

B. 设置文本的字号有两种办法：一种是设置绝对字号，另一种是设置文本的相对字号

C. "+"号表示字体变小

D. "−"号表示字体变大

10. 下列对于滚动字符说法错误的是(　　)。

A. 在 HTML 页面中可以用<marquee></marquee>标记对实现文字的滚动

B. color 属性用于设定活动字幕的背景颜色

C. Direction 属性用于设定活动字幕的滚动方向

D. scrolldelay 属性用于设定两次滚动之间的延迟时间

11. 编写一个数学试卷网页，具体要求如下：

- 试卷名称是标题 2(h2)，每道题的题干是标题 4(h4)；
- 请使用下画线标出正确答案；
- 每道题的内容在网页中占独立一段(p)，段和段之间有个水平线(hr)；
- 网页运行效果如图 3.17 所示。

12. 编写一个网页，网页中有三段滚动的文字，具体要求如下：

- 一段文字滚动方向为向上，文字滚动的背景色为红色，文字滚动 5 次就停止滚动；
- 第二段文字滚动方向为向右；
- 第三段文字滚动的背景色为灰色，文字滚动方式为来回交替滚动；
- 没有指明的属性值就是默认的，网页运行效果如图 3.18 所示。

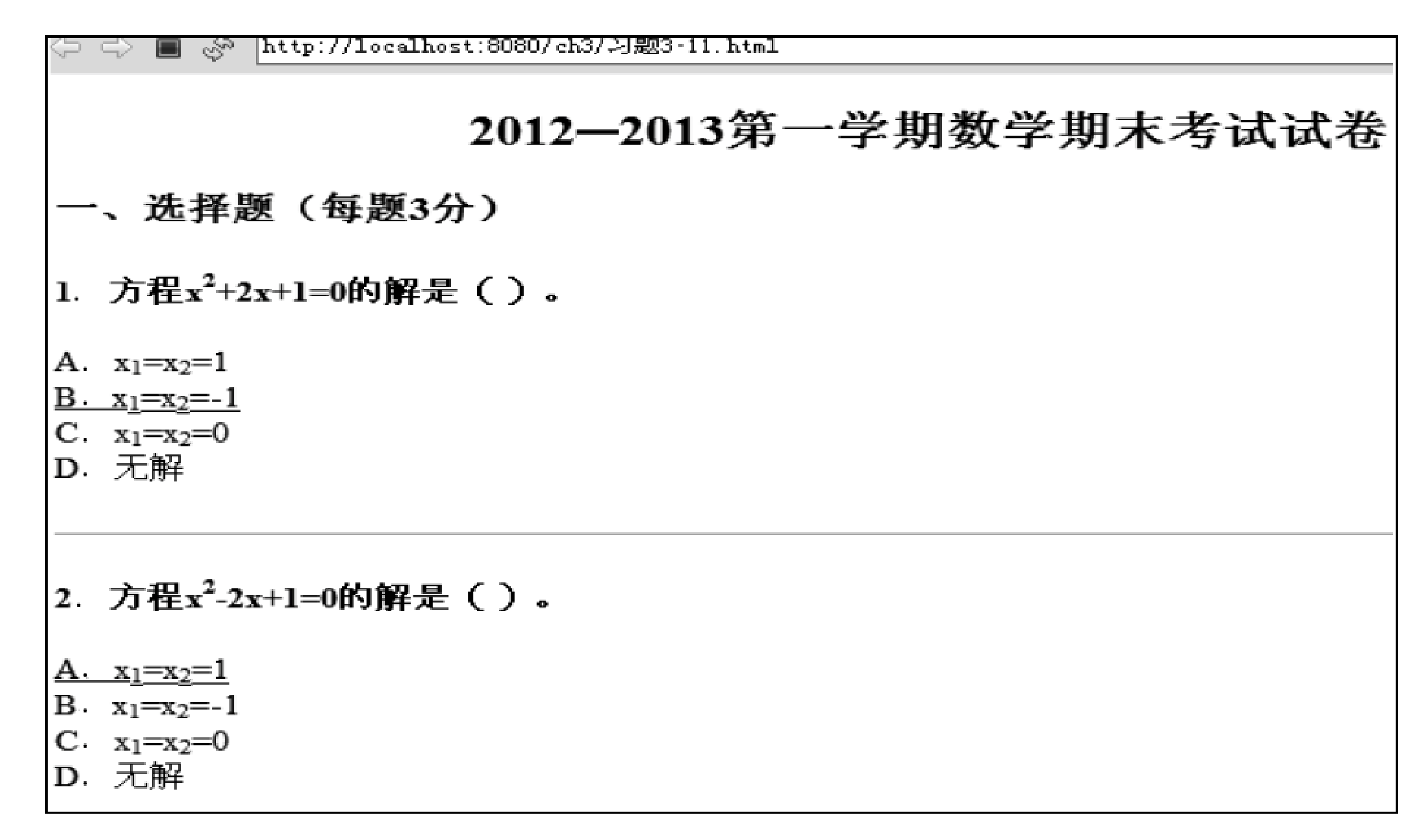


图 3.17　习题 3-11 网页运行效果

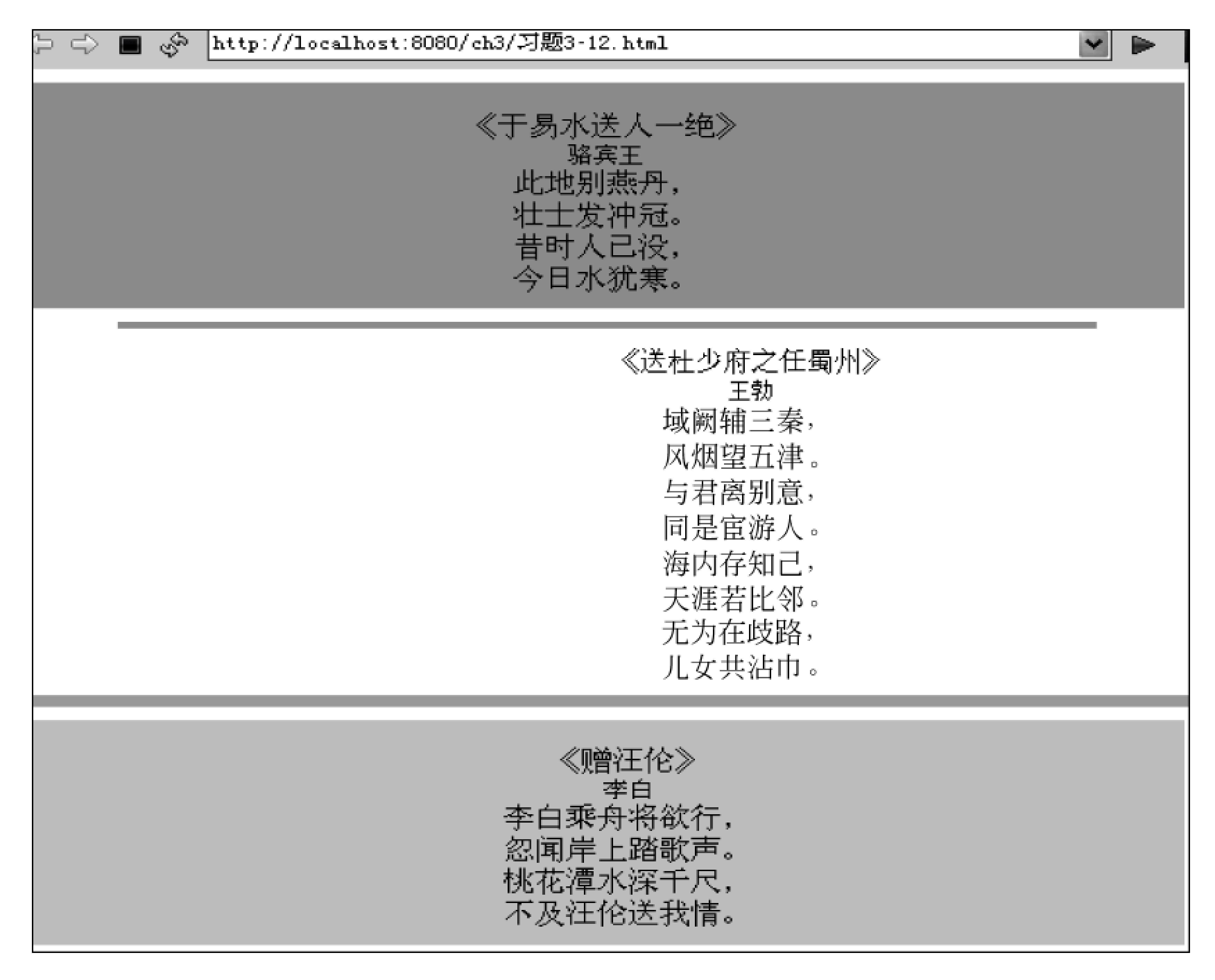


图 3.18　习题 3-12 网页运行效果

第 4 章

列　　表

主要内容

- 无序列表
- 有序列表
- 嵌套序列表

在一个网页中,列表用来设置结构化的、易读的文本格式,可以帮助用户方便地找到信息,并引起用户对信息的注意。本章将介绍各种不同类型的列表,包括无序列表、有序列表以及多级列表。

4.1 无序列表

4.1.1 核心知识

1. 无序列表标记 ul

ul 标记用于设置无序列表,在每个列表项目文字之前,以项目符号作为每条列表项的前缀,各个列表没有级别之分。无序列表语法格式如下:

```
<ul>
    <li>列表项</li>
    <li>列表项</li>
    …
</ul>
```

其中,
表示无序列表的开始,
表示无序列表的结束,
表示一个列表项的开始,
表示一个列表项的结束。

2. 无序列表的项目符号

无序列表的项目符号,默认情况下是"•",而通过 ul 标记的 type 属性可以改变无序列表的项目符号,避免项目符号的单调,语法格式如下:

```
<ul type="符号类型">
    <li>列表项</li>
    <li>列表项</li>
```

```
    …
</ul>
```

其中,type 属性值决定了列表项目的符号。当 type 属性值为 disc 时,项目符号为“•”;当 type 属性值为 circle 时,项目符号为“◦”;当 type 属性值为 square 时,项目符号为“▪”。

3. 定义列表标记 dl

dl 标记用于名词解释,包括定义名称和定义内容。定义列表由<dl>开始,由</dl>结尾。<dt>用来指定定义名称,<dd>用来指定定义内容。定义列表语法格式如下:

```
<dl>
    <dt>定义名称</dt>
    <dd>定义内容</dd>
    <dt>定义名称</dt>
    <dd>定义内容</dd>
    …
</dl>
```

其中,<dl>标记定义了定义列表的开始,</dl>定义了定义列表的结束。<dt>后面就是要解释的名称,<dd>后面则添加该名词的具体解释。

4.1.2 能力目标

能够使用 ul 标记设计无序列表,并会使用 type 属性设置项目符号;能够使用 dl 标记设计定义列表。

4.1.3 任务驱动

1. 任务的主要内容

编写一个网页(效果如图 4.1 所示),在网页中定义一个无序列表,列表项目符号为“◦”。另外,该网页中还有个定义列表,定义名称是“俱乐部”。

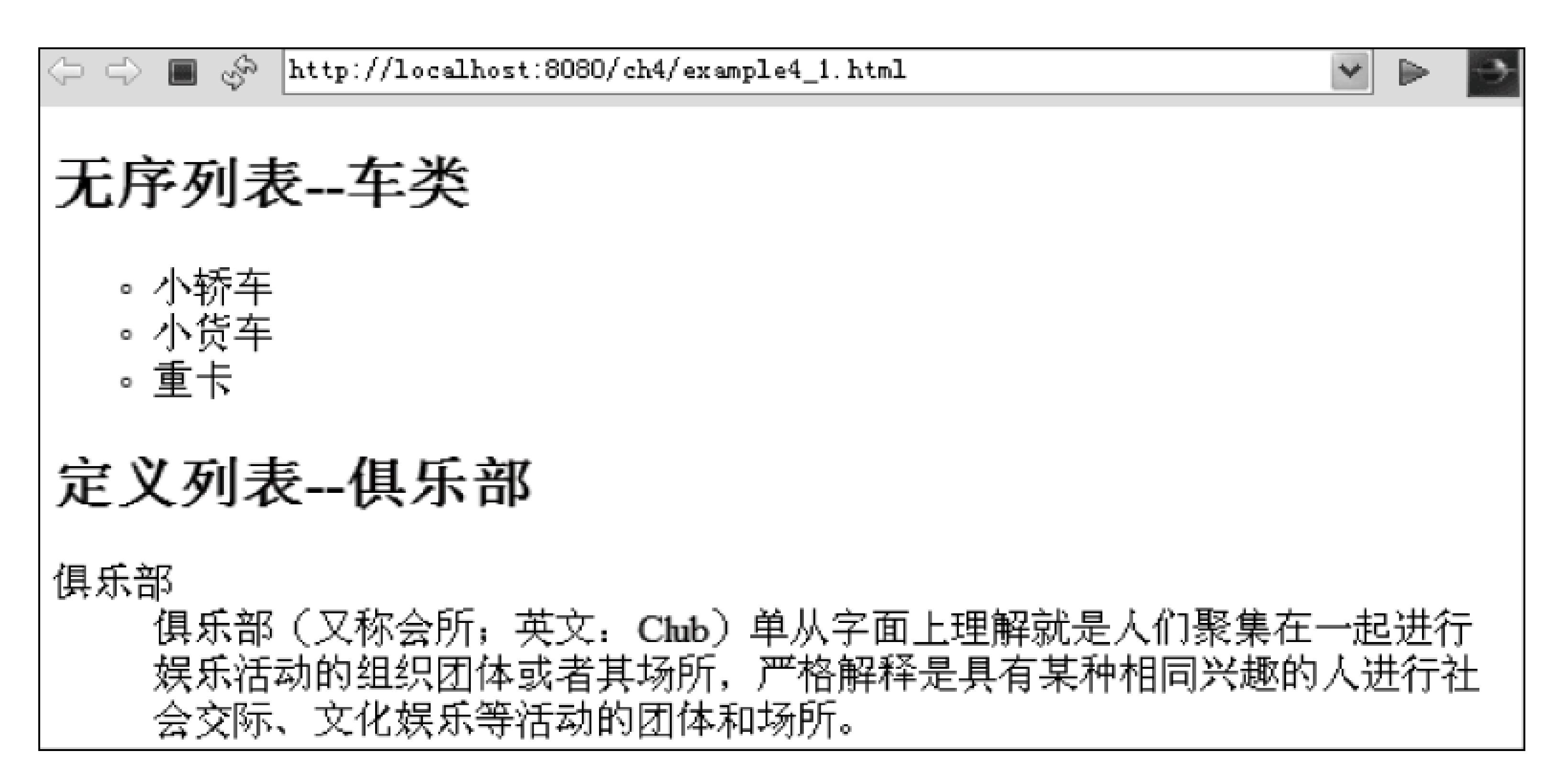


图 4.1 example4_1.html 运行效果

2. 任务的代码模板

将下列 example4_1.html 中的【代码】替换为 HTML 代码。

example4_1. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example4_1.html</title>
</head>
<body>
    <h2>无序列表--车类</h2>
    <ul【代码 1】><!--设置列表项目符号为“◦” -->
        <li>小轿车</li>
        <li>小货车</li>
        <li>重卡</li>
    </ul>
    <h2>定义列表--俱乐部</h2>
    【代码 2】<!--定义列表开始 -->
        <dt>俱乐部</dt>
        <dd>俱乐部(又称会所;英文:Club)单从字面上理解就是人们聚集在一起进行娱乐活动的
            组织团体或者其场所,严格解释是具有某种相同兴趣的人进行社会交际、文化娱乐等
            活动的团体和场所。</dd>
    【代码 3】<!--定义列表结束 -->
</body>
</html>
```

3. 任务小结或知识扩展

从任务中可以看出定义列表的项目是没有项目符号的。另外,定义列表的标记<dl>首尾必须成对出现,而<dt>和<dd>的结尾标记是可以省略的。

4. 代码模板的参考答案

```
【代码 1】: type="circle"
【代码 2】: <dl>
【代码 3】: </dl>
```

4.1.4 实践环节

把 example4_1. html 中的【代码 1】替换成“type="square"”,【代码 2】替换成“<dl type="circle">”,然后运行 example4_1. html 网页,并分析网页的显示效果。

4.2 有 序 列 表

4.2.1 核心知识

1. 有序列表标记 ol

有序列表中的项目采用数字或英文字母开头,通常各项目之间是有先后顺序的。有序列表语法格式如下:

```
<ol>
    <li>列表项</li>
    <li>列表项</li>
    …
</ol>
```

其中，
表示有序列表的开始，
表示有序列表的结束，
表示一个列表项的开始，
表示一个列表项的结束。

2. 有序列表的项目类型

有序列表同无序列表一样，也有项目类型，也可以通过 type 属性设置自己的项目类型。默认情况下，有序列表的项目序号是数字，语法格式如下：

```
<ol type="序号类型">
    <li>列表项</li>
    <li>列表项</li>
    …
</ol>
```

其中，type 属性决定项目序号的类型。当 type 属性值为 1 时，项目序号为“1、2、3、4…”；当 type 属性值为 a 时，项目序号为“a、b、c、d…”；当 type 属性值为 A 时，项目序号为“A、B、C、D…”；当 type 属性值为 i 时，项目序号为“ⅰ、ⅱ、ⅲ、ⅳ…”；当 type 属性值为Ⅰ时，项目序号为“Ⅰ、Ⅱ、Ⅲ、Ⅳ…”。

3. 有序列表的起始数值

默认情况下，有序列表的序号是从 1 开始的，但可以通过 start 属性改变序号的起始值，语法格式如下：

```
<ol start="起始数值">
    <li>列表项</li>
    <li>列表项</li>
    …
</ol>
```

其中，起始数值只能是数字，但是同样对字母或罗马数字起作用。例如，项目类型为 a，起始值为 5，那么项目序号就从英文字母 e 开始编号。

4.2.2 能力目标

能够使用 ol 标记设计有序列表，会使用 type 属性设置项目序号；能够使用 start 属性设定序号的起始值。

4.2.3 任务驱动

1. 任务的主要内容

编写一个网页，具体要求如下：

- 网页中有个<h2>标题：“计算机网络专业的学生应该具备的能力”；
- 网页中有个有序列表描述各项能力，列表项目序号为“1、2、3、4…”；

- 网页运行效果如图 4.2 所示。

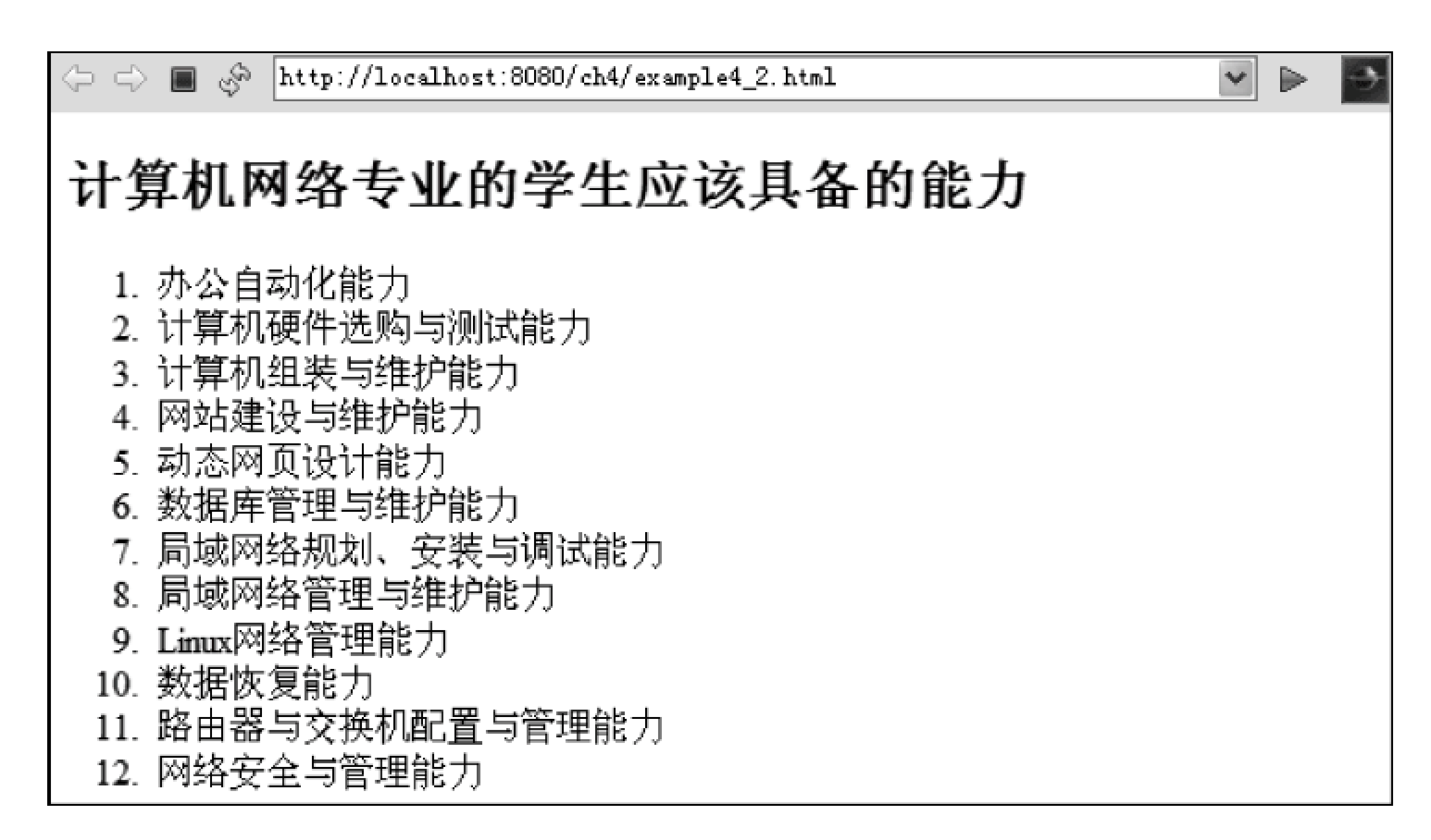


图 4.2　example4_2.html 运行效果

2. 任务的代码模板

将下列 example4_2.html 中的【代码】替换为 HTML 代码。

example4_2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example4_2.html</title>
</head>
<body>
    【代码 1】<!--设置 h2 标题“计算机网络专业的学生应该具备的能力”-->
    【代码 2】<!--设置有序列表并且项目符号为“1、2、3、4…” -->
        <li>办公自动化能力</li>
        <li>计算机硬件选购与测试能力</li>
        <li>计算机组装与维护能力</li>
        <li>网站建设与维护能力</li>
        <li>动态网页设计能力</li>
        <li>数据库管理与维护能力</li>
        <li>局域网络规划、安装与调试能力</li>
        <li>局域网络管理与维护能力</li>
        <li>Linux 网络管理能力</li>
        <li>数据恢复能力</li>
        <li>路由器与交换机配置与管理能力</li>
        <li>网络安全与管理能力</li>
    【代码 3】<!--有序列表结束 -->
</body>
</html>
```

3. 任务小结或知识扩展

如果有序列表的项目编号是罗马数字，那么编号一般控制在 20 以内。如果罗马数字较

大，不仅需要调整显示空间，而且读者很难直观地把罗马数字同阿拉伯数字联系起来。

4. 代码模板的参考答案

【代码 1】：<h2>计算机网络专业的学生应该具备的能力</h2>
【代码 2】：<ol type="1">
【代码 3】：

4.2.4 实践环节

编写一个网页，具体要求如下：

- 网页中有个<h2>标题："计算机网络专业的学生应该学习的课程"；
- 网页中有个有序列表描述所学习的课程，列表项目编号为"A、B、C、D…"，但项目编号从 E 开始；
- 网页运行效果如图 4.3 所示。

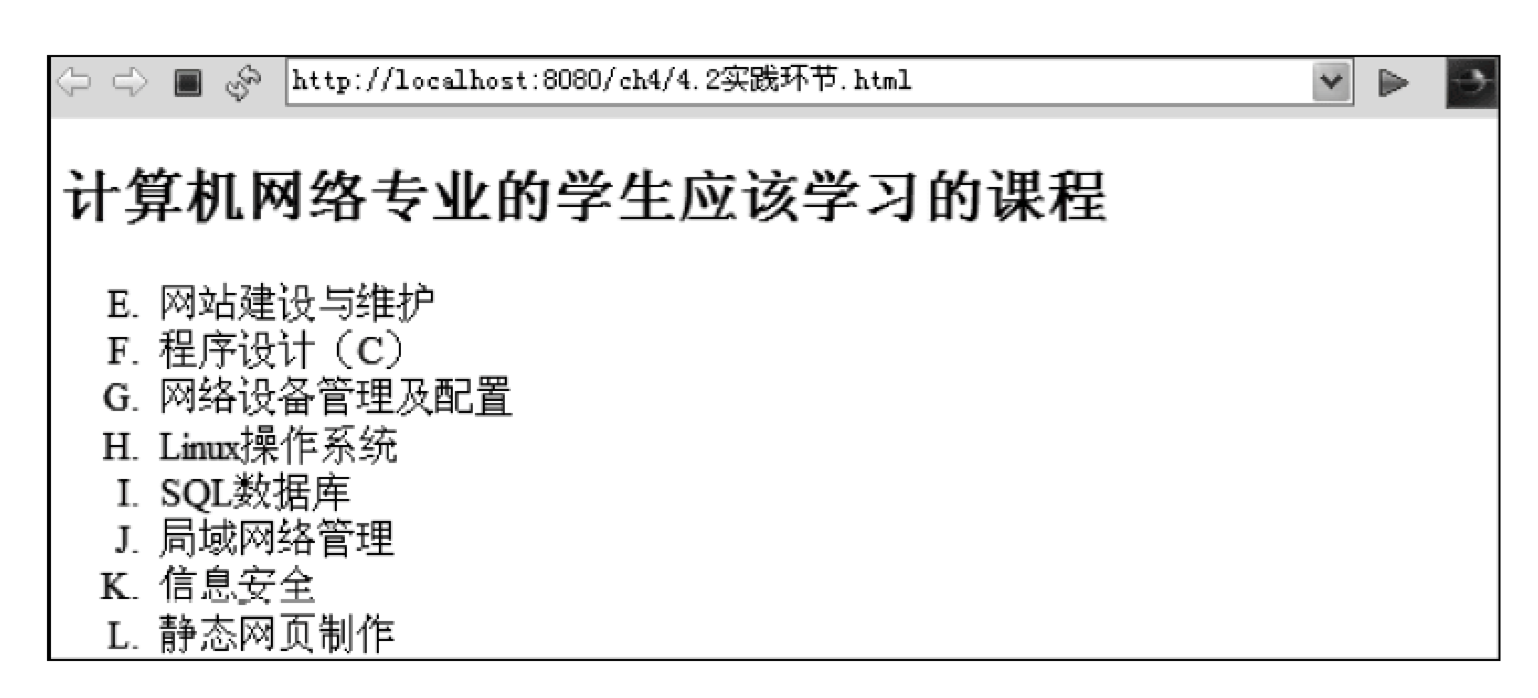


图 4.3　4.2 实践环节.html 运行效果

4.3　嵌套序列表

4.3.1 核心知识

无序列表和有序列表的嵌套是常见的列表嵌套，多次使用<ol>和<ul>标记就可以组合出多种嵌套形式，语法格式如下：

```
<ul>
    <li>名词一
    <ol>
        <li>列表项
        <li>列表项
        …
    </ol>
    <li>名词二
    <ol>
        <li>列表项
        <li>列表项
        …
```

```
    </ol>
    …
</ul>
```

432　能力目标

能够使用无序列表标记<ul>和有序列表标记<ol>设计嵌套列表。

433　任务驱动

1. 任务的主要内容

编写一个网页，具体要求如下：

- 网页中有个无序列表，列表项目符号为“•”；
- 无序列表中共有两个列表项目，分别是“爱情的定义”和“爱情的分类”；
- “爱情的定义”列表项目中嵌套一个有序列表，该有序列表是对爱情进行定义；
- “爱情的定义”列表项目中的有序列表的项目编号为“1、2、3、4…”；
- “爱情的分类”列表项目中嵌套一个有序列表，该有序列表是对爱情进行分类；
- “爱情的分类”列表项目中的有序列表的项目编号为“Ⅰ、Ⅱ、Ⅲ、Ⅳ…”；
- 网页中的文字出自网站 http://baike.baidu.com/view/1348.htm；
- 网页运行效果如图 4.4 所示。

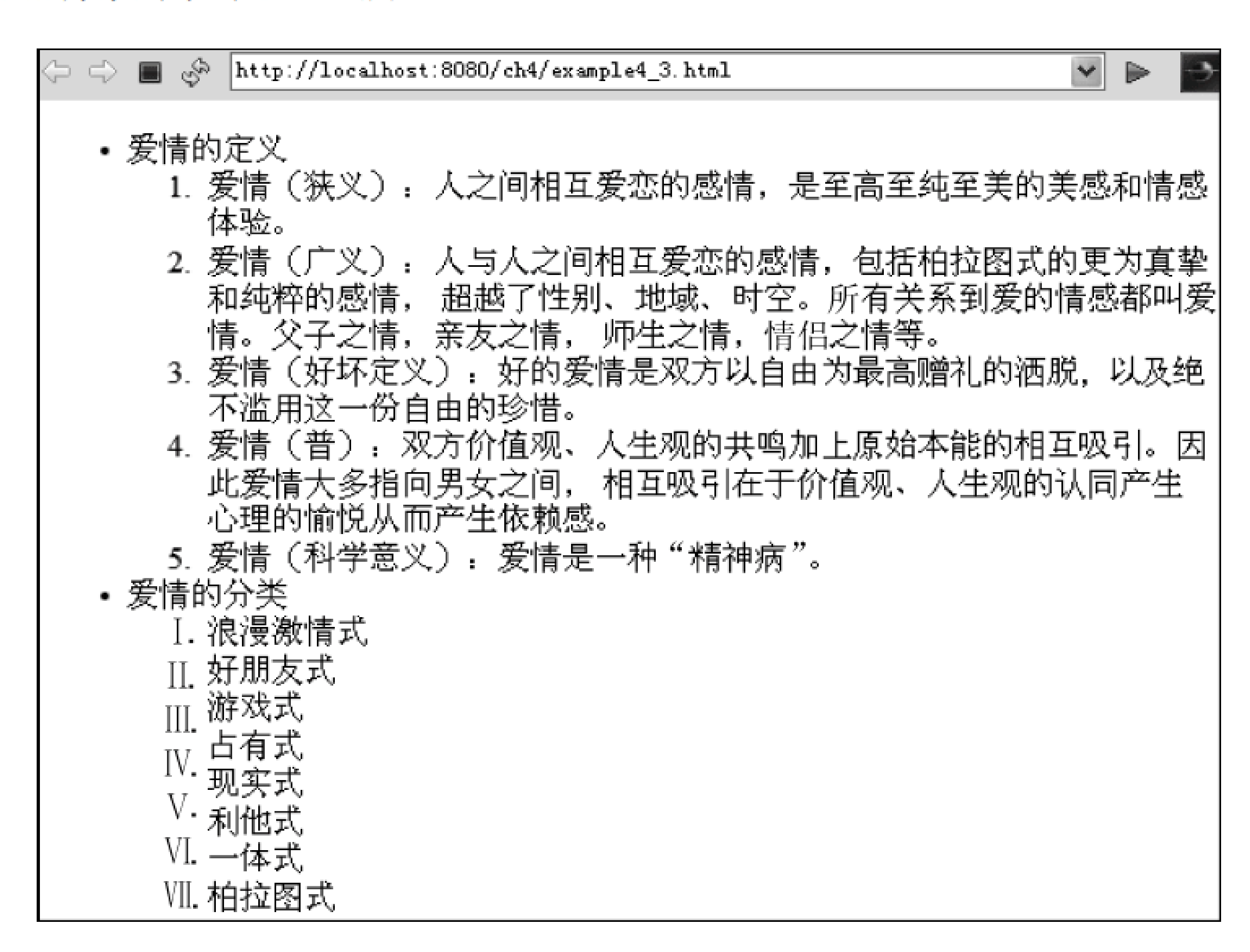


图 4.4　example4_3.html 运行效果

2. 任务的代码模板

将下列 example4_3.html 中的【代码】替换为 HTML 代码。

example4_3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example4_3.html</title>
</head>
<body>
    <ul>
        【代码 1】<!--设置列表项目“爱情的定义”-->
            【代码 2】<!--设置有序列表 ol 的开始标记-->
                <li>爱情(狭义):人之间相互爱恋的感情,是至高至纯至美的美感和情感体验。
                    </li>
                <li>爱情(广义):人与人之间相互爱恋的感情,包括柏拉图式的更为真挚和纯粹
                    的感情,超越了性别、地域、时空。所有关系到爱的情感都叫爱情。父子之
                    情,亲友之情,师生之情,情侣之情等。</li>
                <li>爱情(好坏定义):好的爱情是双方以自由为最高赠礼的洒脱,以及绝不滥用
                    这一份自由的珍惜。</li>
                <li>爱情(普):双方价值观、人生观的共鸣加上原始本能的相互吸引。因此爱情
                    大多指向男女之间,相互吸引在于价值观、人生观的认同产生心理的愉悦从
                    而产生依赖感。</li>
                <li>爱情(科学意义):爱情是一种“精神病”。</li>
            </ol>
        【代码 3】<!--设置列表项目“爱情的分类”-->
            【代码 4】<!--设置有序列表并且项目符号为“Ⅰ、Ⅱ、Ⅲ、Ⅳ…”-->
                <li>浪漫激情式</li>
                <li>好朋友式</li>
                <li>游戏式</li>
                <li>占有式</li>
                <li>现实式</li>
                <li>利他式</li>
                <li>一体式</li>
                <li>柏拉图式</li>
            </ol>
    </ul>
</body>
</html>
```

3. 任务小结或知识扩展

列表的嵌套可以有多种形式:无序列表中嵌套有序列表,无序列表中嵌套无序列表,有序列表中嵌套无序列表,有序列表中嵌套有序列表。

在嵌套列表中,包含其他标记的标记称之为父标记,通常被嵌套的标记要缩进,这样容易看出它们与父标记之间的关系。

4. 代码模板的参考答案

```
【代码 1】:<li>爱情的定义</li>
【代码 2】:<ol>
【代码 3】:<li>爱情的分类</li>
【代码 4】:<ol type="I">
```

434　实践环节

编写一个网页，具体要求如下：

- 网页中有个<h2>标题："调查问卷"；
- 网页中有 3 个有序列表，其中两个有序列表嵌套在另一有序列表中；
- 最外层的有序列表的项目序号为"1、2、3、4…"；
- 被嵌套的两个有序列表的项目序号为"A、B、C、D…"；
- 网页运行效果如图 4.5 所示。

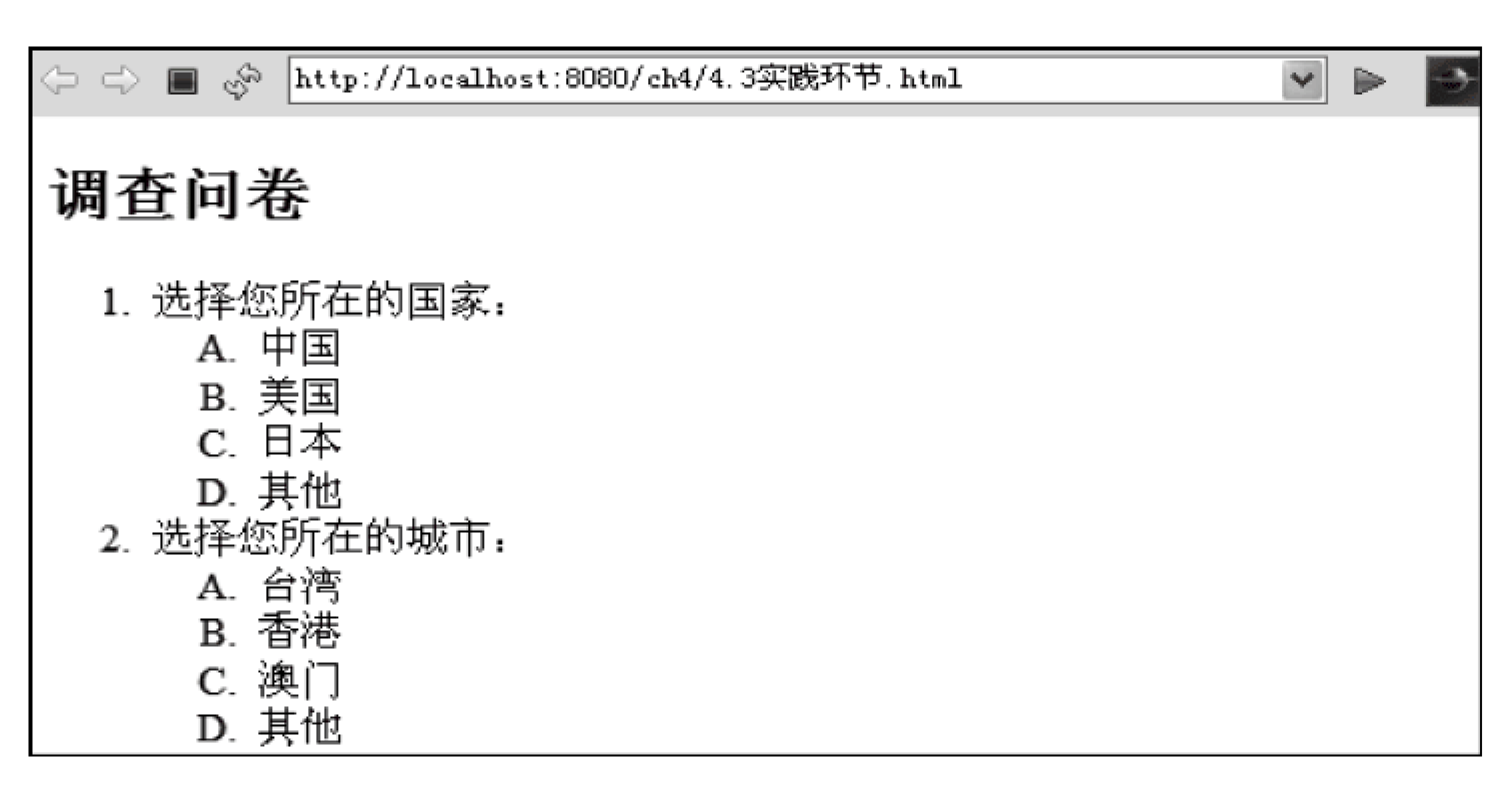


图 4.5　4.3 实践环节.html 运行效果

4.4　小　　结

ul 标记用于设置无序列表，无序列表的项目符号默认是"•"，但通过 type 属性值可以改变无序列表的项目符号。当 type 属性值为 disc 时，项目符号为"•"；当 type 属性值为 circle 时，项目符号为"◦"；当 type 属性值为 square 时，项目符号为"▪"。

ol 标记用于设置有序列表，有序列表的项目序号默认是阿拉伯数字，但通过 type 属性值可以改变有序列表的项目序号。当 type 属性值为 1 时，项目序号为"1、2、3、4…"；当 type 属性值为 a 时，项目序号为"a、b、c、d…"；当 type 属性值为 A 时，项目序号为"A、B、C、D…"；当 type 属性值为 i 时，项目序号为"ⅰ、ⅱ、ⅲ、ⅳ…"；当 type 属性值为 Ⅰ 时，项目序号为"Ⅰ、Ⅱ、Ⅲ、Ⅳ…"。

列表可以任意地嵌套：无序列表中嵌套有序列表，无序列表中嵌套无序列表，有序列表中嵌套无序列表，有序列表中嵌套有序列表。

习　题　4

1. 下列对于清单的说法错误的是(　　)。

   A. 常用的列表有 3 种格式，即无序列表(unordered List)，有序列表(ordered list)和定义列表(definition list)

   B. 无序列表用<ul>开始

C. 无序列表输出时,每一条目前都只能有一个黑色圆点

D. 各种列表可以相互嵌套

2. 编写一个网页,具体要求如下:

- 网页中有个<h2>标题:“个人餐饮调查问卷”;
- 网页中有个有序列表,该有序列表有 2 个列表项目,分别是“你喜欢的海鲜类菜谱:”和“你喜欢的饮料类:”,列表项目序号为“1、2、3、4…”;
- “你喜欢的海鲜类菜谱:”列表项目中嵌套一个无序列表,该无序列表的项目符号为“▪”;
- “你喜欢的饮料类:”列表项目中嵌套一个无序列表,该无序列表的项目符号为“◦”;
- 网页运行效果如图 4.6 所示。

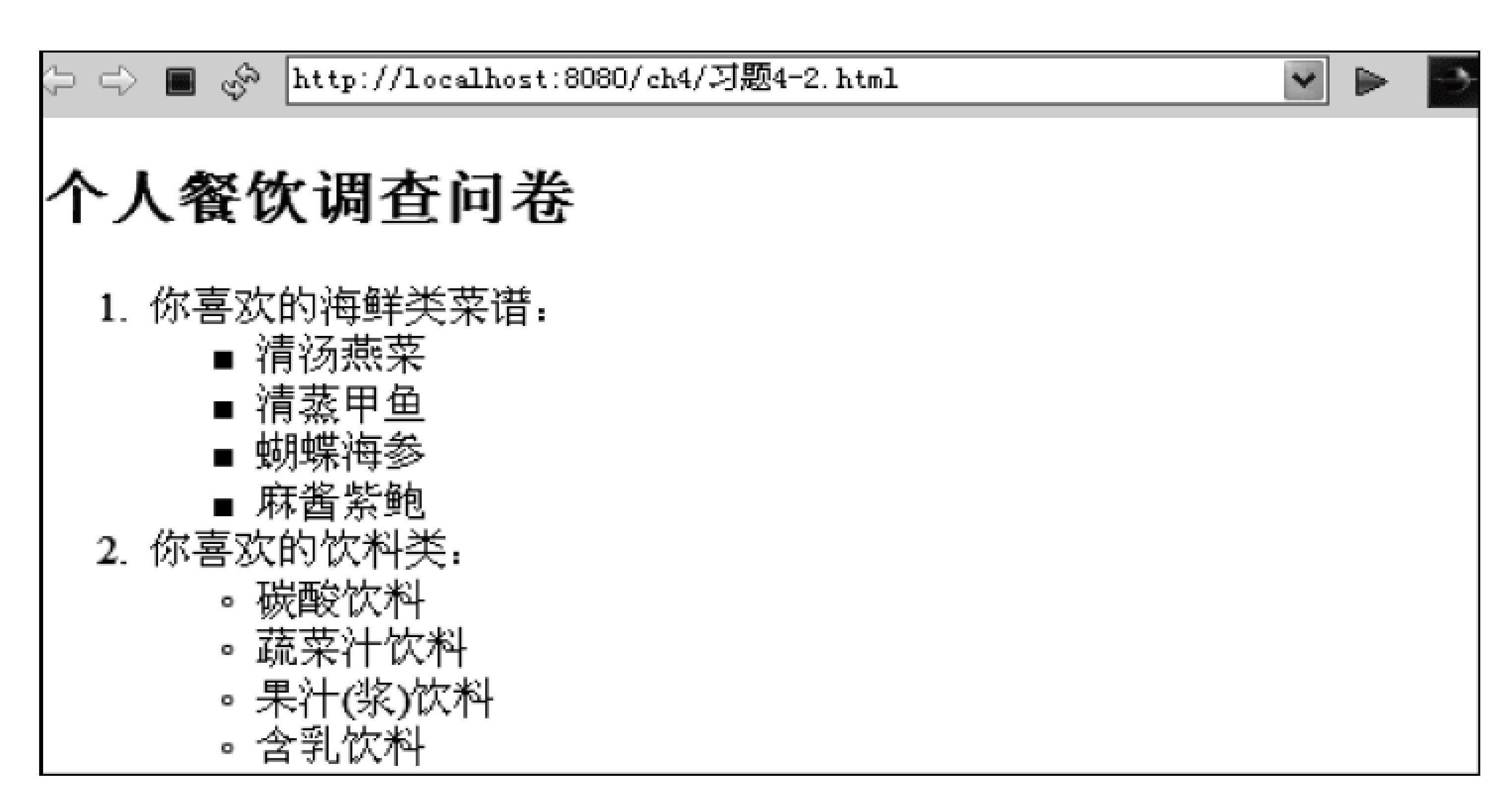


图 4.6　习题 4-2.html 运行效果

第 5 章

图像与多媒体

主要内容

- 网页中使用图像
- 添加网页背景音乐
- 网页中使用多媒体

图像和多媒体在一个网页中起到美化的作用,有了图像和多媒体后,网页不再是单调的文字。本章主要介绍在网页中如何使用图像、多媒体(如动画、声音以及视频)以及添加背景音乐。

5.1 在网页中使用图像

5.1.1 核心知识

在网页中使用图像可以使网页生动,内容更加清晰,起到美化网页的作用。但是,图像过多会造成网页图像下载时间过长,会使浏览者失去耐心。一般情况下,要用最少的字节数生成高质量的图像。网页中常用的图像格式有 JPEG、GIF、BMP、PCX、TIFF 和 PNG 等。

1. 标记 img

在网页中使用图像时,需要使用 img 标记插入图像。img 标记有很多属性,其中 src 属性是必需的,它指定要插入图像文件的位置与名称,语法格式如下:

```
<img src="图像文件的路径及名称">
```

其中,图像文件的路径可以是相对路径,也可以是绝对路径。

2. img 标记的相关属性

1) alt 属性

alt 属性用于设置提示文字,提示文字会在图像无法显示时替代图像在浏览器中显示,语法格式如下:

```
<img src="图像文件的路径及名称" alt="提示文字内容">
```

其中,提示文字内容只要是浏览器能识别的文字就可以。

2）height 与 width 属性

默认情况下,网页中图像的显示大小就是图像的宽度和高度,但可以使用 height 和 width 属性任意设置图像显示的高度和宽度,语法格式如下:

```
<img src="图像文件的路径及名称" width="图像的宽度" height="图像的高度">
```

其中,图像高度和宽度的单位是像素。

3）border 属性

默认情况下,网页中的图像是没有边框的,如果要设置图像的边框可以使用 border 属性。语法格式如下:

```
<img src="图像文件的路径及名称" border="图像边框的宽度">
```

在上述语法中,边框的单位是像素,宽度值越大边框越宽。

4）hspace 与 vspace 属性

网页中图像与文字之间的水平距离可以通过 hspace 属性设置,垂直距离可以通过 vspace 属性设置。语法格式如下:

```
<img src="图像文件的路径及名称" hspace="水平间距" vspace ="垂直间距">
```

其中,水平间距与垂直间距的单位是像素。

5）align 属性

可以使用 align 属性设置图像的对齐方式,语法格式如下:

```
<img src="图像文件的路径及名称" align ="对齐方式">
```

其中,align 属性值常用的有 3 种:middle(居中)、left(居左)、right(居右)。

5.1.2 能力目标

能使用 img 标记在网页中添加图像,并且能使用 img 标记的各种属性设置图像在网页中对应的属性值。

5.1.3 任务驱动

1. 任务的主要内容

编写一个网页(运行效果如图 5.1 所示),具体要求如下:

- 网页中有个<h2>标题:"雄猫战斗机";
- 标题的下面有张战斗机图片 plane.jpg,图片文件和网页文件在同一目录中;
- 图像的宽度为 500,图像的高度为 400;
- 图像的边框宽度为 2;
- 图像的提示文字设置为:"俺是美国的高级战斗机"。

2. 任务的代码模板

将下列 example5_1.html 中的【代码】替换为 HTML 代码。



图 5.1　example5_1.html 运行效果

example5_1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example5_1.html</title>
</head>
<body>
<center>
    <h2>雄猫战斗机</h2>
    雄猫战斗机<【代码 1】src="plane.jpg"【代码 2】="500"【代码 3】="400"【代码 4】="2"
    【代码 5】="俺是美国的高级战斗机">是美国 F-14 舰载战斗机。
<!--代码 1 使用 img 插入图片,代码 2 设置图片的宽度,代码 3 设置图片的高度,代码 4 设置图片的
边框,代码 5 设置图片的提示文字 -->
</center>
</body>
</html>
```

3. 任务小结或知识扩展

一般情况下,通过程序把图像的尺寸放大会产生马赛克效果,显得粗糙,而缩小就不会有这样的问题。如果图像尺寸确实太小,那么最好使用图像处理软件(如 Photoshop CS)重新生成一张尺寸合适的图像,然后再插入到网页中。

4. 代码模板的参考答案

【代码 1】: img
【代码 2】: width
【代码 3】: height
【代码 4】: border
【代码 5】: alt

5.1.4 实践环节

编写一个网页,具体要求如下:

- 网页中有个<h3>标题:“漂亮的星海湾”;
- 标题的下面有张风景图片 xinghaiwan.jpg,图片文件和网页文件在同一目录中;
- 图像的宽度为 200,图像的高度为 200;
- 图像与文字的水平间距为 20,垂直间距也为 20;
- 图像的对齐方式为居中;
- 网页运行效果如图 5.2 所示。

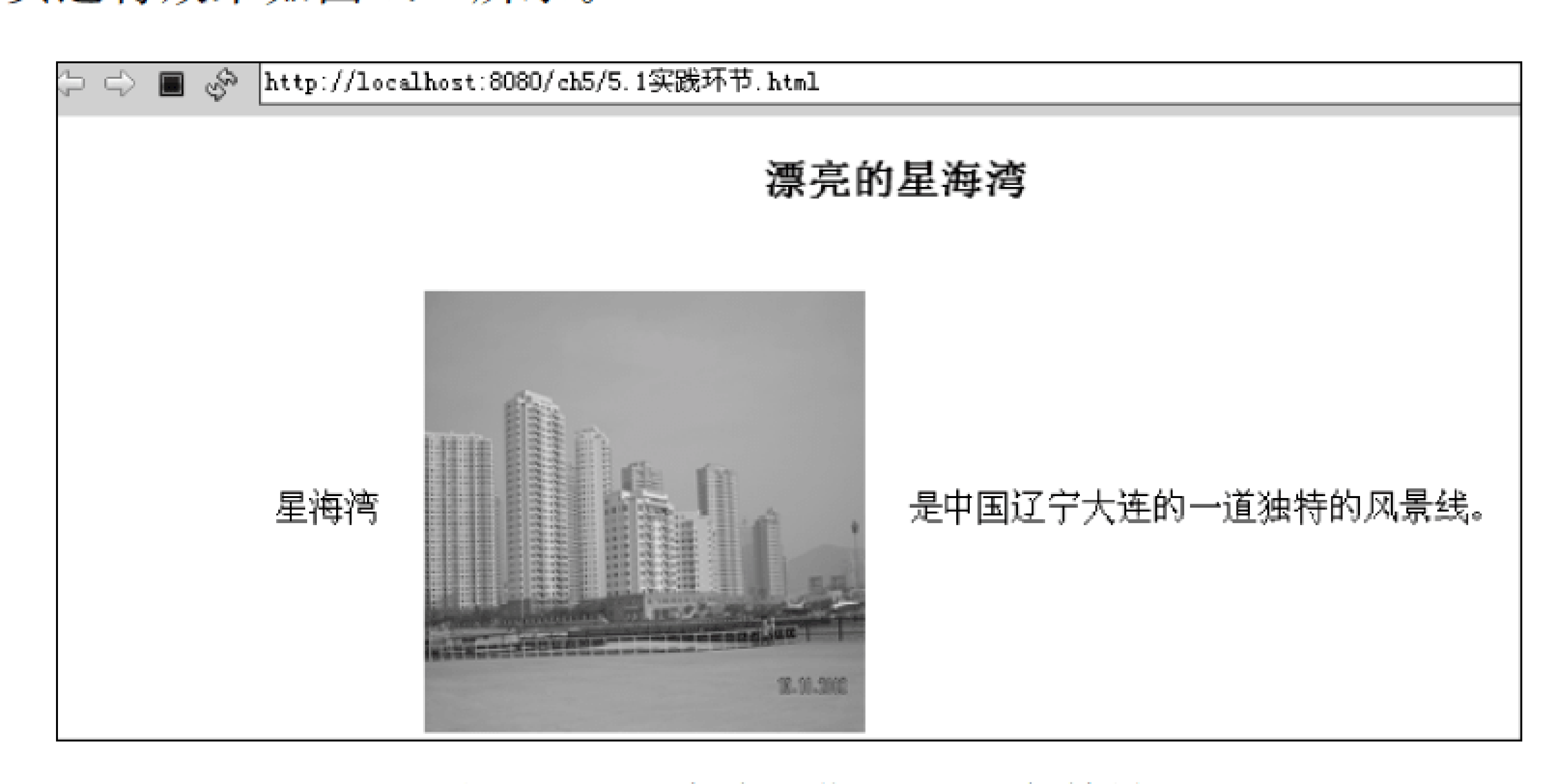


图 5.2　5.1 实践环节.html 运行效果

5.2　添加网页背景音乐

5.2.1 核心知识

给网页添加背景音乐,可以使浏览者在进入网站的同时能听到优美的音乐,也可以大大增强网站的娱乐效果。使用 bgsound 标记给网页添加背景音乐,语法格式如下:

```
<bgsound src="音乐文件的路径及名称" loop="播放次数">
```

其中,音乐文件的格式可以是 AVI、MP3 以及 MID 等类型的声音文件。如果希望背景音乐无限次地循环播放,那么 loop 的值设置为 true。

5.2.2 能力目标

能使用 bgsound 标记给网页添加背景音乐,以增强网页的娱乐效果。

5.2.3　任务驱动

1. 任务的主要内容

编写一个网页(运行效果如图 5.3 所示),网页中有个<h1>标题:“边听着优美的歌声,边看着紧张的叙利亚新闻”,还有个<h2>标题:“叙利亚谴责联合国人权理事会决议不公”。网页有背景音乐,背景音乐文件(传奇.mp3)与网页的程序文件在同一个目录中,背景音乐循环播放两次结束。

http://localhost:8080/ch5/example5_2.html

边听着优美的歌声，边看着紧张的叙利亚新闻

叙利亚谴责联合国人权理事会决议不公

新华网大马士革12月3日电（记者李震）叙利亚外交部3日强烈谴责联合国 人权理事会日前通过的决议不公正，强调该决议完全是基于敌对势力的造谣煽动和虚假信息。

官方的叙利亚通讯社当天援引外交部的一份声明说，联合人权理事会的决议案是“耸人听闻和不公正的”，是叙利亚敌对势力早有预谋的，目的就是要让叙利亚问题政治化。

声明强调，决议是根据叙利亚境外敌对势力和不诚实的媒体机构散布的“造谣煽动和不实信息”。叙利亚政府一直积极致力于维护国家的安全与稳定，保护人民的利益，努力推动改革，决议粗暴干涉了叙利亚的内政。

联合国人权理事会2日在日内瓦召开叙利亚问题特别会议，以37票赞成，4票反对，6票弃权通过了一项由欧盟提出有关叙利亚人权形势的决议。

决议对叙利亚问题独立国际调查委员会提交的调查报告表示欢迎；强烈谴责叙利亚政府“大规模、有组织侵犯人权和基本自由的行为”；敦促叙利亚政府停止一切侵犯人权做法，释放被捕人士，接受国际监督。

图 5.3　example5_2.html 运行效果

2. 任务的代码模板

将下列 example5_2.html 中的【代码】替换为 HTML 代码。

example5_2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example5_2.html</title>
</head>
<body>
    <h1 align="center">边听着优美的歌声,边看着紧张的叙利亚新闻</h1>
    <h2 align="center">叙利亚谴责联合国人权理事会决议不公</h2>
    <!--代码 1 使用 bgsound 插入背景音乐,代码 2 使用 loop 设置背景音乐播放的次数 -->
    <【代码 1】="传奇.mp3"【代码 2】>
    <p>
        新华网大马士革 12 月 3 日电(记者李震)叙利亚外交部 3 日强烈谴责联合国人权理事会日
        前通过的决议不公正,强调该决议完全是基于敌对势力的造谣煽动和虚假信息。
    <p>
```

```
        官方的叙利亚通讯社当天援引外交部的一份声明说,联合人权理事会的决议案是“耸人听闻
        和不公正的”,是叙利亚敌对势力早有预谋的,目的就是要让叙利亚问题政治化。
    <p>
        声明强调,决议是根据叙利亚境外敌对势力和不诚实的媒体机构散布的“造谣煽动和不实信
        息”。叙利亚政府一直积极致力于维护国家的安全与稳定,保护人民的利益,努力推动改革,
        决议粗暴干涉了叙利亚的内政。
    <p>
        联合国人权理事会2日在日内瓦召开叙利亚问题特别会议,以37票赞成,4票反对,6票弃
        权通过了一项由欧盟提出有关叙利亚人权形势的决议。
    <p>
        决议对叙利亚问题独立国际调查委员会提交的调查报告表示欢迎;强烈谴责叙利亚政府
        “大规模、有组织侵犯人权和基本自由的行为”;敦促叙利亚政府停止一切侵犯人权做法,释
        放被捕人士,接受国际监督。
</body>
</html>
```

3. 任务小结或知识扩展

有的浏览者喜欢背景音乐,有的浏览者不喜欢背景音乐。对于 bgsound 标记添加的背景音乐,浏览者是无法控制的。因此,在给网页添加背景音乐时,要慎重。

4. 代码模板的参考答案

【代码1】: bgsound src
【代码2】: loop="2"

5.2.4 实践环节

编写一个网页(运行效果如图5.4所示),具体要求如下:



图5.4　5.2实践环节.html运行效果

- 网页中有个<h2>标题："此网页既有背景图片，又有背景音乐"；
- 背景图片文件(plane.jpg)和背景音乐文件(传奇.mp3)与网页的程序文件在同一个目录中；
- 背景音乐无限制地循环播放。

5.3　在网页中使用多媒体

5.3.1　核心知识

如何使单调的网页变得生动呢？那可能离不开音频或视频文件。浏览者要想正确地浏览这些被添加视频文件的网页，必须在自己的计算机上安装相应的播放软件。

1. 多媒体文件标记 embed

在网页中可以使用标记 embed 将多媒体文件(如 Flash 动画、MP3 音乐、ASF 视频等)添加到网页中，语法格式如下：

```
<embed src="多媒体文件的路径及名称" width="播放器的宽度" height="播放器的高度">
</ embed>
```

其中，多媒体文件的路径可以是相对路径，也可以是绝对路径。播放器的宽度和高度一定要设置，否则无法显示播放多媒体文件的软件。

2. 使用 autostart 属性设置自动运行

使用 autostart 属性设置多媒体文件自动运行，语法格式如下：

```
<embed src="多媒体文件的路径及名称" width="播放器的宽度" height="播放器的高度"
autostart="是否自动运行"></embed>
```

其中，autostart 的取值有两个，一个是 false，表示不自动播放多媒体文件；一个是 true，表示自动播放多媒体文件。

3. 使用 loop 属性设置循环播放

```
<embed src="多媒体文件的路径及名称" width="播放器的宽度" height="播放器的高度"
loop="是否循环播放"></embed>
```

其中，loop 的取值有两个，一个是 false，表示只播放一次多媒体文件；一个是 true，表示无限制地重复播放多媒体文件。

4. 使用 hidden 属性设置播放器的隐藏

```
<embed src="多媒体文件的路径及名称" width="播放器的宽度" height="播放器的高度" hidden="
是否显示播放器"></embed>
```

其中，hidden 的取值有两个，一个是 false，表示显示播放器；一个是 true，表示隐藏播放器。

5.3.2　能力目标

能使用 embed 标记给网页添加各种多媒体文件(如 Flash 动画、MP3 音乐、ASF 视频

等)，使网页效果更加生动。

5.3.3 任务驱动

1. 任务的主要内容

编写一个网页(运行效果如图 5.5 所示)，具体要求如下：

- 网页中有个<h2>标题："我们毕业了"；
- 网页中有 3 段文字，3 段文字之后有个 Flash 动画(我们毕业了.swf)；
- Flash 动画文件和网页程序文件在同一个目录中；
- Flash 动画播放器的高度和宽度都是 500；
- Flash 动画文件不自动播放。



图 5.5 example5_3.html 运行效果

2. 任务的代码模板

将下列 example5_3.html 中的【代码】替换为 HTML 代码。

example5_3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example5_3.html</title>
</head>
<body>
```

```
    <h2 align="center">我们毕业了</h2>
    <p>
        今天,我们毕业了,从此,我们告别了一段纯真的青春,一段年少轻狂的岁月,一个充满幻想
        的时代……
    <p>
        毕业前的这些日子,时间过得好像流沙,看起来漫长,却无时无刻不在逝去;想挽留,一伸手,
        有限的时光却在指间悄然溜走,毕业答辩,散伙席筵,举手话别,各奔东西……一切似乎都预
        想的到,一切又走得太过无奈。
    <p>
        这几天,我们都会有意无意地再逛逛校园,看一看它今天的样子,想一想五年前它如何迎来
        稚气未脱的我们。走了五年,似乎又走回了起点。突然觉得,五年的同窗、身边的朋友,比想
        象中要和善、可爱得多!星光下的夜晚,每一个都温柔如风。
<br>
<center>
    <!--代码 1 使用 embed 插入 Flash 动画,代码 2 使用 autostart 设置不自动播放 -->
    <【代码 1】="我们毕业了.swf" width="500" height="500"【代码 2】></embed>
</center>
</body>
</html>
```

3. 任务小结或知识扩展

多媒体固然能使网页产生美好的效果,但是使用多媒体时,要注意浏览器是否集成了多媒体播放软件。

4. 代码模板的参考答案

【代码 1】: embed src
【代码 2】: autostart="false"

5.3.4 实践环节

编写一个网页(运行效果如图 5.6 所示),具体要求如下:

- 网页中有个<h3>标题:“请您欣赏音乐”;



图 5.6 5.3 实践环节.html 运行效果

- 网页中使用多媒体文件(醉清风.mp3)；
- MP3 文件和网页程序文件在同一个目录中；
- MP3 播放器的高度和宽度都是 200；
- MP3 播放器打开网页时，自动播放 MP3 文件；
- MP3 播放器不停地循环播放 MP3 文件。

5.4 小　　结

使用 img 标记可以在网页中插入图像将网页变得生动美观，但大量的图像会影响网页的运行速度。因此，应先用最小的字节数生成高质量的图像，然后再插入到网页中。

使用 bgsound 标记可以给网页添加背景音乐，但有的网页浏览者并不希望听到背景音乐，自己还没有办法把背景音乐去掉。因此，网页制作者要慎重添加背景音乐。

使用 embed 标记可以将多媒体文件(如 Flash 动画、MP3 音乐、ASF 视频等)添加到网页中，浏览者要想正确地浏览这些被添加多媒体文件的网页，必须在自己的计算机上安装相应的播放软件。

习　题　5

1. 常用的网页图像格式有(　　)和(　　)。

   A. gif,tiff　　B. tiff,jpg　　C. gif,jpg　　D. jpeg,png

2. 有关网页中图像的说法不正确的是(　　)。

   A. 网页中的图像并不与网页保存在同一个文件中，每个图像单独保存

   B. HTML 语言可以描述图像的位置、大小等属性

   C. HTML 语言可以直接描述图像上的像素

   D. 图像可以作为超链接的起始对象

3. 在网页中若要播放名为 demo.avi 的动画，以下选项中正确的是(　　)。

   A. <Embed src="demo.avi" autostart=true>

   B. <Embed src="demo.avi" autoopen=true>

   C. <Embed src="demo.avi" autoopen=true></Embed>

   D. <Embed src="demo.avi" autostart=true></Embed>

4. 若要循环播放背景音乐 bg.mid，以下选项中正确的是(　　)。

   A. <bgsound src="bg.mid" Loop="1">

   B. <bgsound src="bg.mid" Loop="True">

   C. <sound src="bg.mid" Loop="True">

   D. <Embed src="bg.mid" autostart="true"></Embed>

5. 下列关于插入图片说法错误的是(　　)。

   A. <img>标记是用来处理图像输出的

   B. “src”属性指明了所要链接图像的文件地址，这个图形文件必须是本地计算机上的图形

C. “width＝＃ 、height＝＃ ”属性用来设定图片大小

D. “hspace＝＃ 、vspace＝＃ ”属性用来设定图片边沿空白，以免文字或其他图片贴近

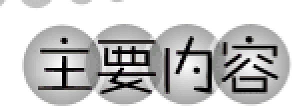
6. 编写一个网页（运行效果如图 5.7 所示），网页中有个＜h2＞标题：“请您慢慢欣赏”。标题的下面有张图片 cat.jpg，图片文件和网页程序在同一目录中。图像的宽度为 100，图像的高度为 100，图像的边框宽度为 1，图像的提示文字设置为：“俺是来搞笑的。”网页中使用多媒体文件（传奇.mp3），MP3 文件和网页程序文件在同一个目录中。MP3 播放器的高度和宽度都是 100，MP3 播放器打开网页时，自动播放 MP3 文件，MP3 播放器不停地循环播放 MP3 文件。



图 5.7　习题 5-6.html 运行效果

第 6 章

超　链　接



- 基本超链接
- 锚点超链接
- 外部网站、E-mail 等其他链接

网站中的一个个网页可以通过超链接关联在一起。因此，超链接是一个网站的灵魂，是网页中最重要的元素之一。本章将介绍基本超链接、锚点链接、外部网站链接以及 E-mail 链接等。

6.1　在网页中使用基本超链接

6.1.1　核心知识

超链接的作用是建立从一个位置到另一个位置的链接。利用超链接不仅可以进行网页间的相互访问，还可以使网页链接到其他相关的多媒体文件上。

1. 超链接标记<a>

超链接标记<a>是一个非常重要的标记，它可以成对出现在文档的任何位置，语法格式如下：

```
<a href="链接路径">链接内容</a>
```

其中，“链接内容”可以是文字内容，也可以是一张图片。

2. 使用 target 属性设置超链接的目标窗口

默认情况下，超链接的目标网页在当前页面中打开(关闭当前页面窗口)，但可以使用 target 属性控制被打开的目标窗口，语法格式如下：

```
<a href="链接路径" target="目标窗口的打开方式">链接内容</a>
```

其中，target 属性值可以为_self、_blank、_top 以及_parent，_self 是 target 的默认值。如果 target 的值为_blank，那么目标页面会在一个新的空白窗口中打开。如果 target 的值为_top，那么目标页面会在顶层框架中打开。如果 target 的值为_parent，那么目标页面会在当

前框架的上一层打开。框架的概念将在第 9 章介绍。

6.1.2　能力目标

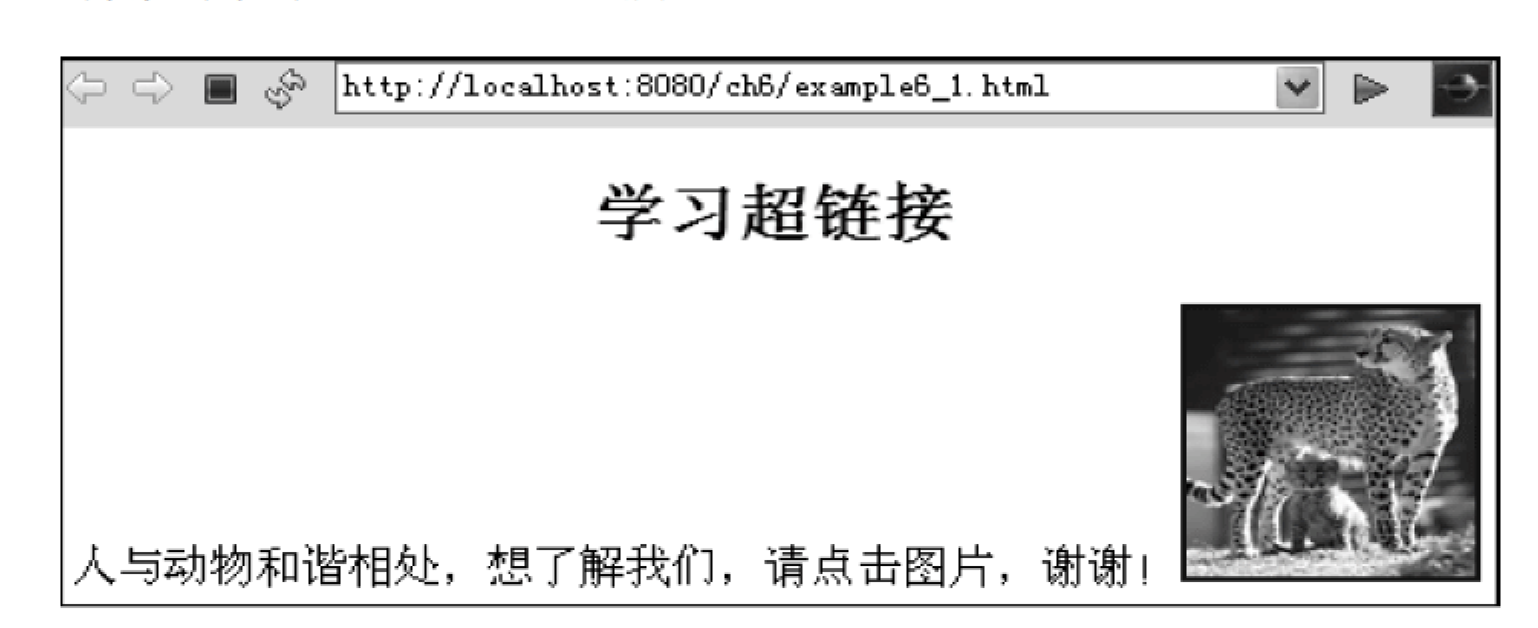
要求读者能使用超链接标记<a>创建链接，并且会使用 target 属性设置超链接目标页面的打开方式。

6.1.3　任务驱动

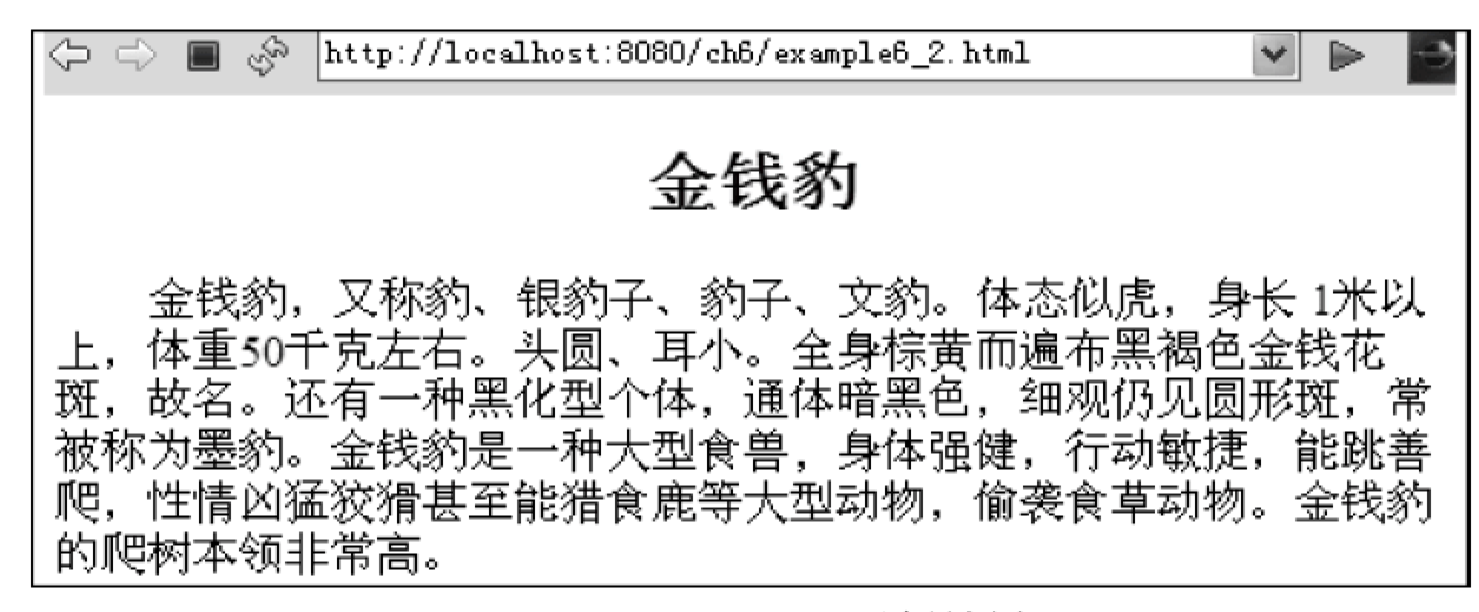
1. 任务的主要内容

编写两个网页 example6_1.html 与 example6_2.html，具体要求如下：

- 在 example6_1.html 页面中有个<h2>标题"学习超链接"。
- 在标题"学习超链接"下面有段文字"人与动物和谐相处，想了解我们，请点击图片，谢谢！"，在该段文字后面有个超链接，超链接的内容为一张图片 cat_bao.jpg（图片与 example6_1.html 页面在同一个目录下），链接路径为"example6_2. html"。
- 在 example6_2.html 页面中有个<h2>标题"金钱豹"。
- 在标题"金钱豹"下面有段文字"金钱豹，又称豹、银豹子、豹子、文豹。体态似虎，身长 1 米以上，体重 50 千克左右。头圆、耳小。全身棕黄而遍布黑褐色金钱花斑，故名。还有一种黑化型个体，通体暗黑色，细观仍见圆形斑，常被称为墨豹。金钱豹是一种大型食兽，身体强健，行动敏捷，能跳善爬，性情凶猛狡猾甚至能猎食鹿等大型动物，偷袭食草动物。金钱豹的爬树本领非常高"。
- 网页运行效果如图 6.1(a)、(b)所示。



(a) example6_1.html运行效果



(b) example6_2.html运行效果

图　6.1

2. 任务的代码模板

将下列 example6_1. html 中的【代码】替换为 HTML 代码。

example6_1. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example6_1.html</title>
</head>
<body>
    <h2 align="center">学习超链接</h2>
    <!--代码 1 使用标记 a 建立链接,目标页面为 example6_2.html -->
    <p>人与动物和谐相处,想了解我们,请点击图片,谢谢!【代码 1】<img src="cat_bao.jpg" width
       ="100" height="100" alt="我们可爱吗?"></a></p>
</body>
</html>
```

example6_2. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example6_2.html</title>
</head>
<body>
    <h2 align="center">学习超链接</h2>
    <p>人与动物和谐相处,想了解我们,请点击图片,谢谢!<a href="example6_2.html">
       <img src="cat_bao.jpg" width="100" height="100" alt="我们可爱吗?"></a>
       </p>
</body>
</html>
```

3. 任务小结或知识扩展

链接路径可以分为相对路径和绝对路径。绝对路径是完全路径,也就是说不管源文件在什么位置,都可以找到它。相对路径是指相对当前文件的简化路径,当当前文件与所链接的文档处于同一个目录内时,相对路径就显得特别方便。

链接路径使用相对路径还是绝对路径,有一条通用的规则:链接目标页面与当前页面在一起时应使用相对路径,链接目标页面在其他地方(其他计算机、硬盘或网站)时,应使用绝对路径。

4. 代码模板的参考答案

【代码 1】: <a href="example6_2.html">

6.1.4　实践环节

编写两个网页(6.1 实践环节_1.html,6.1 实践环节_2.html),具体要求如下:

- 在 6.1 实践环节_1.html 中有个<h2>标题:“个人简介”;
- “个人简介”标题的下面有 4 段文字;
- 在 6.1 实践环节_1.html 的最后一行有个超链接(个人相册),链接目标页面为 6.1 实践环节_2.html,目标页面在新的窗口中打开;
- 在 6.1 实践环节_2.html 中使用 img 标记插入 3 张图片,图片与网页文件在同一个目录中,图片宽度都是 352,图片高度都是 470;
- 网页中用到的文字与照片出自网站 http://baike.baidu.com/view/2101049.htm;
- 6.1 实践环节_1.html 运行效果如图 6.2(a)所示,6.1 实践环节_2.html 运行效果如图 6.2(b)所示。



顾小白，自2000年起开始撰写电影随笔及评论，曾为《看电影》、《电影世界》、《21世纪环球报道》、《信息时报》、《青年时报》、《花溪》、《瑞丽》、新浪文化频道、网易娱乐频道、tom娱乐频道等媒体撰写专栏，参撰书籍包括《家卫森林》、《独立精神》、《电影行走》、《后窗看电影》等。

2003年出版个人随笔集《失忆谅解备忘录》（现代出版社），2005年5月最新推出《顾小白电影随笔——等待是一生最初苍老》（古吴轩出版社）。

2002年完成二十集电视剧处女作《捆绑上天堂》（李少红导演，因故未拍），2006年完成电影编剧处女作《红色康拜因》（蔡尚君导演），该片获第十二届釜山国际电影节费比西国际影评人奖及希腊国际电影节最佳影片金亚历山大奖。

2008年参与李少红版《红楼梦》的编剧，2010年和《贱人》的作者尹丽川、《轮廓回暖》的作者多多共同参与张艺谋导演的新戏《山楂树之恋》的电影创作。

个人相册

(a) 6.1实践环节_1.html运行效果



(b) 6.1实践环节_2.html运行效果

图　6.2

6.2 在网页中使用锚点链接

6.2.1 核心知识

浏览网页的时候，如果网页的内容比较多会导致页面过长，需要不断地拖拉滚动条才能看完网页的所有内容，这样很不方便。如何避免这样的情况呢？那就是需要在网页中使用锚点链接。

1. 锚点的创建

锚点与链接的文字可以在同一个页面，也可以在不同的页面。在实现锚点链接之前，需要先创建锚点，通过创建的锚点才能对页面的内容进行引导与跳转。创建锚点的语法格式如下：

```
<a name="锚点的名称"></a>
```

其中，锚点的名称可以是数字或英文字母，或者两者混合。在同一页面中可以有多个锚点，但名称不能相同。

2. 在同一页面中使用锚点

一个锚点创建后，就可以在同一页面内使用锚点链接了，使用锚点链接的语法格式如下：

```
<a href="#锚点的名称"></a>
```

其中，“#”后面写上锚点的名称后，就可以链接到同一个页面中由该锚点指定的位置了，这种方式只是在同一页面中使用锚点。

3. 在不同页面中使用锚点

在不同页面中使用锚点链接，具体语法格式如下：

```
<a href="链接路径#锚点的名称"></a>
```

6.2.2 能力目标

不仅能够在同一个页面中使用锚点链接，而且还能够在不同页面中使用锚点链接。

6.2.3 任务驱动

1. 任务的主要内容

编写两个网页 example6_3.html 与 example6_4.html，具体要求如下：

- 在 example6_3.html 页面中有 4 个超链接：狮子、人类、斑马以及印度豹，它们分别链接到不同的锚点。其中，“狮子”链接到“Lion”锚点，“人类”链接到“example6_4.html#Race”锚点，“斑马”链接到“Zebra”锚点，“印度豹”链接到“example6_4.html#Cheetah”锚点。
- 在 example6_3.html 页面中定义了两个锚点：Lion 与 Zebra，在 example6_4.html 页面中定义了两个锚点：Race 与 Cheetah。
- 在 example6_3.html 与 example6_4.html 页面中使用到的图片与两个页面同属一

个目录。

2. 任务的代码模板

将下列 example6_3. html 与 example6_4. html 中的【代码】替换为 HTML 代码。

example6_3. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>人与动物和谐相处</title>
</head>
<body>
<p><a【代码 1】>狮子</a>          <!--链接到 Lion 锚点 -->
    <br>
<br>
        <a【代码 2】>人类</a>      <!--链接到 example6_4.html#Race 锚点 -->
    <br>
    <br>
        <a【代码 3】>斑马</a>      <!--链接到 Zebra 锚点 -->
    <br>
    <br>
        <a【代码 4】>印度豹</a>  <!--链接到 example6_4.html#Cheetah 锚点 -->
    <br>
    <br>
    <br>
    <br>
        <a【代码 5】>狮子</a>      <!--定义锚点 Lion -->
</p>
<p><img src="44.gif" width="255" height="167"></p>
<p><img src="33.jpg" width="255" height="177"></p>
<p><img src="lion.gif" width="256" height="169"></p>
<p>狮子的吼声从八千米之外就能听到!雄狮(很容易从鬃毛识别出雌雄)的重量高达 250 千克。而
    雌狮则要小得多,重 180 千克。
<br>
<br>
<a【代码 6】>斑马</a>              <!--定义锚点 Zebra -->
<p><img src="66.jpg" width="400" height="269">
<p><img src="77.jpg" width="400" height="266">
<p><img src="55.jpg" width="400" height="269">
<p>没有任何两匹斑马的斑纹完全一样,因此每匹斑马都是独一无二的。斑马也称为黑白条纹相间
    的马。大多数动物学家相信斑纹对动物界的活动有重要作用,即可以通过它来区分斑马与其他
    动物。
</body>
</html>
```

example6_4. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<head>
<title>人与动物和谐相处</title>
</head>
<body>
<a【代码 7】>人类</a><!--定义锚点 Race -->
<p><img src="race.jpg" width="100" height="100">
<p>人种(race),亦称种族,是由具有形态上和生理上的特点和语言习俗等历史文化因素组成的有区
    域性特点的群体。根据人种的自然体质特征,生物学家以本质主义方式(即以体质特征为标准)
    通常将全世界的现代人类划分为四大人种:欧罗巴人种(又称白色人种或高加索人种或欧亚人
    种)、蒙古人种(又称黄色人种或亚美人种)、尼格罗人种(又称黑色人种或赤道人种)和澳大利亚
    人种(又称大洋洲人种或棕色人种),俗称白种人、黄种人、黑种人和棕种人。
<br><br>
<a【代码 8】>印度豹</a>            <!--定义锚点 Cheetah -->
<p><img src="2.jpg" width="311" height="450">
<p><img src="1.jpg" width="410" height="356">
<p>印度豹以快速奔跑著称,其速度可与飞机媲美。它是所有陆上动物中跑得最快的动物。
</body>
</html>
```

3. 任务小结或知识扩展

从上面任务中可以看出,使用锚点链接浏览页面方便了许多,但是如果锚点使用不当,页面跳转会显得非常混乱。

4. 代码模板的参考答案

【代码 1】: href=#Lion
【代码 2】: href=example6_4.html#Race
【代码 3】: href=#Zebra
【代码 4】: href=example6_4.html#Cheetah
【代码 5】: name=Lion
【代码 6】: name=Zebra
【代码 7】: name=Race
【代码 8】: name=Cheetah

6.2.4 实践环节

编写网页(6.2实践环节.html),网页中有个<h2>标题:“吉林长白山”,标题下面一行有5个链接:简介、区域、地理信息、胜景、长白山旅游线路,分别链接到不同的5个锚点上:#jianjie、#quyu、#dili、#shengjing、#lvyou,页面运行效果如图6.3所示。

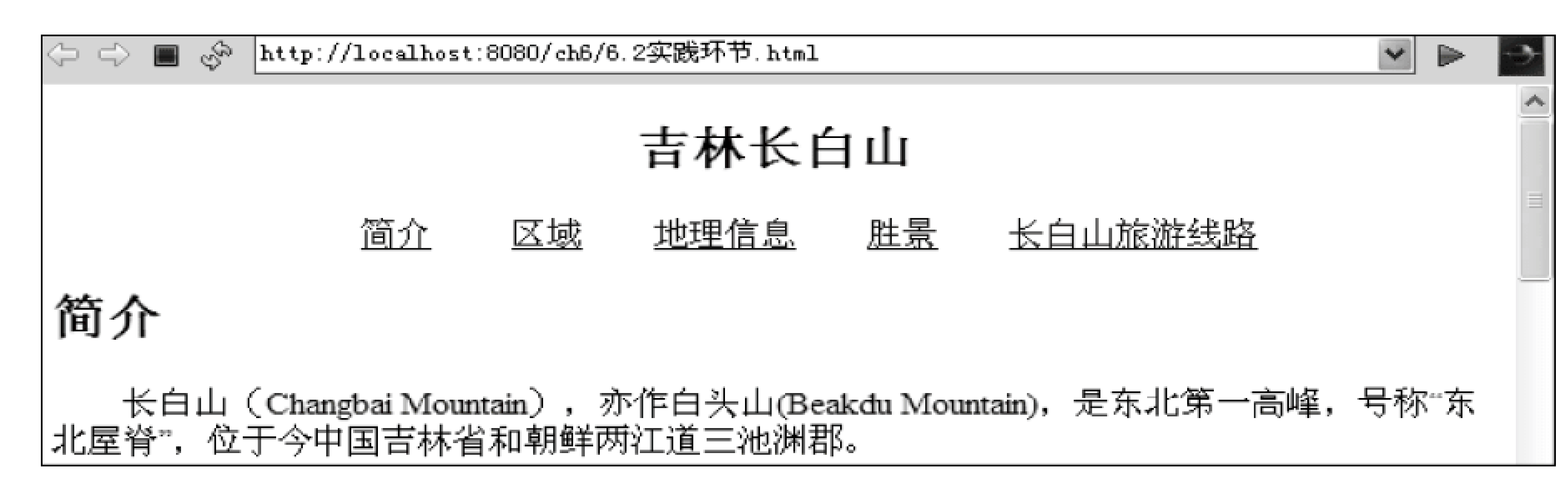


图6.3　6.2实践环节.html运行效果

6.3　在网页中使用其他链接

6.3.1　核心知识

在网页中除了创建以上讲述的链接外，还可以创建到外部网站的链接、下载文件链接以及 E-mail 链接。

1. 外部网站的链接

链接到外部网站是指从当前网站跳转到另一个网站，一般情况下这种链接需要使用绝对的链接路径，大多数使用 HTTP 协议实现外部网站的链接，语法格式如下：

```
<a href="http://网站地址">链接内容</a>
```

其中，“http://”表明有关 HTTP 协议的外部网站的链接，在其后输入网址即可链接到该网站。

2. 下载文件链接

当需要在网站中提供资料下载时，就需要为资料文件提供下载链接。如果超链接指向的不是一个网页文件，而是其他文件，如 ZIP、RAR、MP3、EXE 文件等，单击链接时就会下载相应的文件。下载文件链接的语法格式如下：

```
<a href="文件路径">链接内容</a>
```

3. E-mail 链接

在网页中使用 E-mail 链接，可以使浏览者快速地写邮件反馈自己的意见。当单击 E-mail 链接时，会立即打开浏览器默认的 E-mail 程序，E-mail 链接会自动写入收件人地址，无须浏览者再次输入。E-mail 链接的语法格式如下：

```
<a href="mailto:电子邮件地址">链接内容</a>
```

其中，电子邮件地址的后面还可以添加一些参数实现邮件发送。这些参数可以是 cc、subject、bcc 以及 body 等。参数语法如下：

（1）抄送收件人的地址

```
<a href="mailto:电子邮件地址?cc=抄送收件人的邮件地址">链接内容</a>
```

（2）电子邮件的主题

```
<a href="mailto:电子邮件地址?subject=主题文字">链接内容</a>
```

（3）暗送收件人的地址

```
<a href="mailto:电子邮件地址?bcc=暗送收件人的邮件地址">链接内容</a>
```

（4）电子邮件的内容

```
<a href="mailto:电子邮件地址?body=邮件内容">链接内容</a>
```

6.3.2 能力目标

能使用<a>标记创建从当前网站到另一个网站的链接，能使用<a>标记创建 E-mail 链接。

6.3.3 任务驱动

1. 任务的主要内容

编写网页(example6_5. html)，网页中个 h2 标题："自制 hao123. com"，标题下面一行有 4 个超链接：百度、新浪、搜狐、文件下载，分别链接到不同的目标页面或文件：http://www. baidu. com、http://www. sina. com. cn、http://www. sohu. com、myfile. rar；网页运行效果如图 6. 4(a)所示，单击"文件下载"超链接时，IE 浏览器会打开下载页面，下载页面如图 6. 4(b)所示。



(a) example6_5.html运行效果



(b) 下载页面

图 6. 4

2. 任务的代码模板

将下列 example6_5. html 中的【代码】替换为 HTML 代码。

example6_5. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example6_5.html</title>
</head>
```

```
<body>
    <h2 align="center">自制 hao123.com</h2>
    <a【代码 1】>百度</a>          <!--链接到 http://www.baidu.com -->
    <a【代码 2】>新浪</a>          <!--链接到 http://www.sina.com.cn -->
    <a【代码 3】>搜狐</a>          <!--链接到 http://www.sohu.com -->
    <a【代码 4】>文件下载</a>      <!--链接到 myfile.rar -->
</body>
</html>
```

3. 任务小结或知识扩展

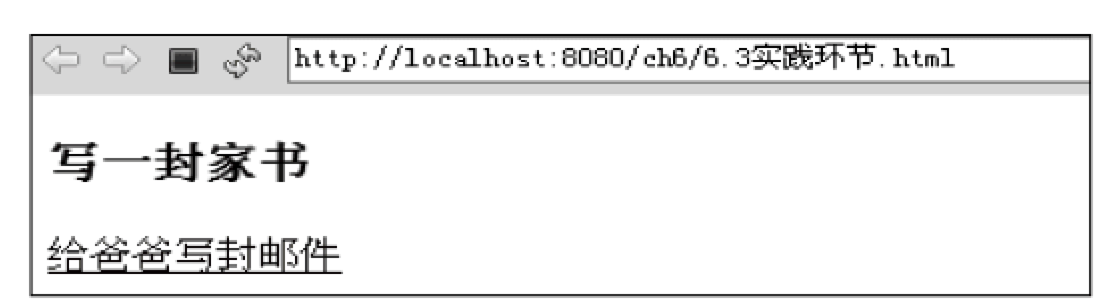
链接到外部网站时，要保障网络是畅通的，否则打不开外部网站的目标页面或文件。

4. 代码模板的参考答案

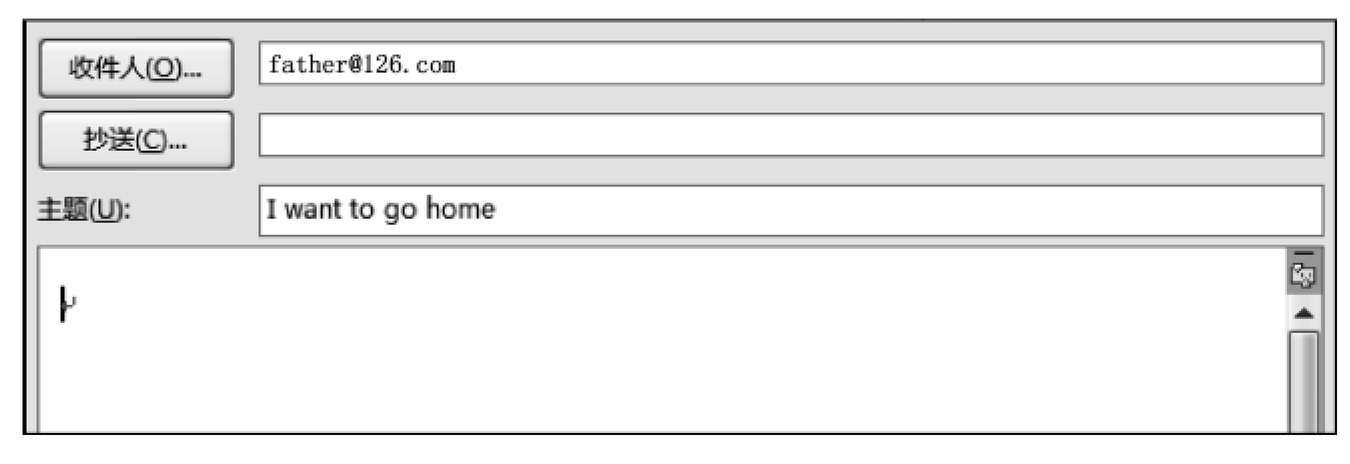
```
【代码 1】: href="http://www.baidu.com"
【代码 2】: href="http://www.sina.com.cn"
【代码 3】: href="http://www.sohu.com"
【代码 4】: href="myfile.rar"
```

6.3.4　实践环节

编写网页(6.3 实践环节.html)，网页中有个 h3 标题："写一封家书"。标题下面一行有个 E-mail 超链接："给爸爸写封邮件"，邮件主题为"I want to go home"，邮件地址为"father@126.com"。网页运行效果如图 6.5(a)所示，单击"给爸爸写封邮件"超链接，会打开 Windows 邮件服务软件 OutLook，如图 6.5(b)所示。



(a) 6.3实践环节.html运行效果



(b) OutLook运行效果

图　6.5

6.4　小　　结

超链接的作用是建立从一个位置到另一个位置的链接。使用<a>标记可以在网页中定义一个超链接，链接路径可以分为相对路径和绝对路径。不管是相对路径还是绝对路径，在使用超链接时，都要把链接路径写正确。

习　题　6

1. 当链接指向下列哪一种文件时，不打开该文件，而是提供给浏览器下载？(　　)

A. ASP　　B. HTML　　C. ZIP　　D. JSP

2. 下列哪一项是在新窗口中打开网页文档？(　　)

A. _self　　B. _blank　　C. _top　　D. _parent

3. 若要在页面中创建一个图像超链接，要显示的图像为 myhome. png，所链接的地址为 http://www. pcnetedu. com，以下选项中正确的是(　　)。

A. myhome. png

B. <img src="myhome. png">

C. <img src="myhome. png">

D. <a href ="http://www. pcneredu. com"><img src="myhome. png">

4. 要将页面的当前位置定义成名为"vbpos"的锚点，其定义方法正确的是(　　)。

A.

B. vbpos

C. <a name="vbpos">

D.

5. 以下创建 E-mail 链接的方法，正确的是(　　)。

A. 管理员

B. 管理员

C. 管理员

D. 管理员

6. 常见的超链接有哪几种？

7. 相对路径与绝对路径有什么区别？什么时候使用相对路径？什么时候使用绝对路径？

第 7 章

表　　格

主要内容

- 表格的基本构成
- 表格的基本属性
- 单元格的基本属性
- 表格的嵌套
- 表格的结构

在一个网页中，无论排列文本内容还是对图像、数据等元素进行排版，表格元素都起到至关重要的作用。表格以简洁明了的方式，将文本内容、数据、图像等元素有序地显示在网页上。本章将重点介绍表格的创建、表格的属性、表格的嵌套以及表格的结构。

7.1　创建表格

7.1.1　核心知识

一个表格由行、列和单元格构成，可以有多行，每行可以有多个单元格。在网页中，表格也是由标记创建的。

1. table、tr 和 td 标记

表格的 3 部分：行、列和单元格，一般通过 3 个标记来创建。标记 table 创建表格，标记 tr 创建行，标记 td 创建单元格。行是水平方式的，贯穿表格的左右；列是垂直的；单元格是行和列交会的部分，是输入和显示信息的地方。创建表格要以＜table＞标记开始，以＜/table＞标记结束，语法格式如下：

```
<table>
    <tr>
        <td>单元格中的内容</td>
        <td>单元格中的内容</td>
        …
    </tr>
    <tr>
        <td>单元格中的内容</td>
```

```
        <td>单元格中的内容</td>
        …
    </tr>
    …
</table>
```

在一个表格中包含几组<tr>和</tr>标记，就表示该表格有几行。<td>和</td>标记分别表示单元格的开始和结束，在一行中包含几组<td>和</td>标记，就表示该行中有几个单元格。

2. 标题标记 caption

一个表格中只含有一个<caption>标记用于设置表格的标题，语法格式如下：

```
<table>
    <caption>表格的标题</caption>
    <tr>
        <td>单元格中的内容</td>
        <td>单元格中的内容</td>
        …
    </tr>
    <tr>
        <td>单元格中的内容</td>
        <td>单元格中的内容</td>
        …
    </tr>
    …
</table>
```

3. 表头标记 th

<th>是<td>单元格的一种变体，实质上是一种单元格，用来显示表头信息。默认情况下，浏览器会以粗体和居中的样式显示<th>标记中的内容。表头标记的语法格式如下：

```
<table>
    <tr>
        <th>表头中的内容</th>
        <th>表头中的内容</th>
        …
    </tr>
    <tr>
        <td>单元格中的内容</td>
        <td>单元格中的内容</td>
        …
    </tr>
    …
</table>
```

7.1.2 能力目标

能使用<table>、<tr>和<td>标记创建表格，会设置一个表格的标题和表头。

7.1.3　任务驱动

1. 任务的主要内容

编写网页 example7_1. html，在该页面中有个表格，表格的标题为“王菲单曲”，表格有 4 行，第一行为表头“序号”与“曲名”，运行效果如图 7.1 所示。

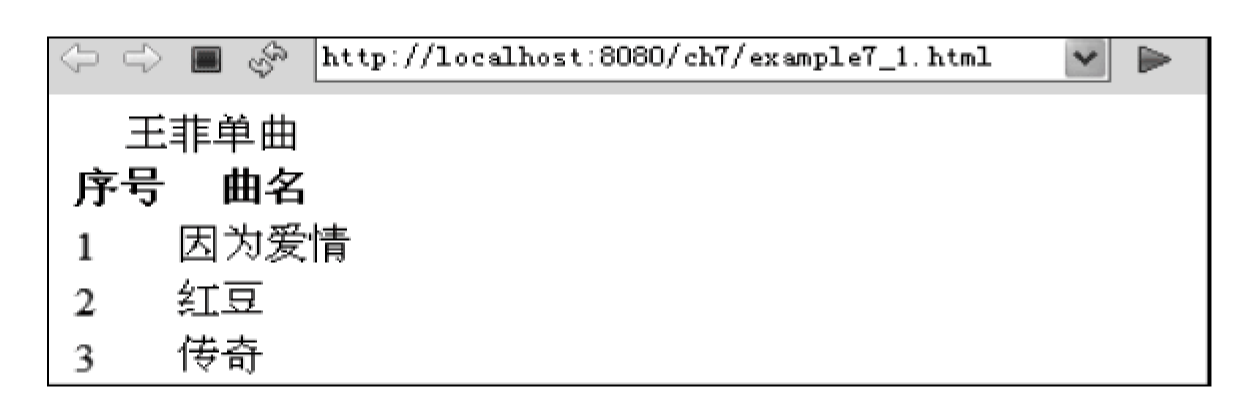


图 7.1　example7_1. html 运行效果

2. 任务的代码模板

将下列 example7_1. html 中的【代码】替换为 HTML 代码。

example7_1. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_1.html</title>
</head>
<body>
    <table>
        【代码 1】<!--创建标题"王菲单曲"-->
        <tr>
            【代码 2】<!--创建表头"序号"-->
            【代码 3】<!--创建表头"曲名"-->
        </tr>
        <tr>
            <td>1</td>
            <td>因为爱情</td>
        </tr>
        <tr>
            <td>2</td>
            <td>红豆</td>
        </tr>
        <tr>
            <td>3</td>
            <td>传奇</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

使用＜caption＞标记创建表格标题的好处是将标题定义在表格中，如果表格移动，则标题会随着表格进行相应的移动。

一般情况下，在表格的第一行使用＜th＞标记设置表头。＜th＞起始标记必须有，＜/th＞结尾标记是可选的。

在一个网页中，根据需要可以使用＜table＞标记创建任意多个表格。

4. 代码模板的参考答案

【代码 1】：<caption>王菲单曲</caption>
【代码 2】：<th>序号</th>
【代码 3】：<th>曲名</th>

7.1.4 实践环节

在 example7_1. html 页面中再次使用＜caption＞标记添加一个标题“caption 在表格中只能使用一次”，运行网页，并分析网页运行效果。

7.2 设置表格的宽度、高度以及对齐方式

7.2.1 核心知识

1. 设置表格的宽度

可以使用 width 属性设置表格的宽度，语法格式如下：

```
<table width="表格宽度">
```

如果不指定表格宽度，则浏览器会根据表格内容自动调整宽度。

2. 设置表格的高度

可以使用 height 属性设置表格的高度，语法格式如下：

```
<table height ="表格高度">
```

如果不指定表格高度，则浏览器会根据表格内容自动调整高度。

3. 设置表格的对齐方式

可以使用 align 属性设置整个表格在页面中的对齐方式，语法格式如下：

```
<table align="对齐方式">
```

默认情况下，整个表格在页面中是左对齐的。当 align 的取值为 left 时，整个表格在页面中左对齐；当 align 的取值为 center 时，整个表格在页面中居中对齐；当 align 的取值为 right 时，整个表格在页面中右对齐。

7.2.2 能力目标

能使用 width、height 和 align 属性设置表格的宽度、高度和对齐方式。

7.2.3　任务驱动

1. 任务的主要内容

编写网页 example7_2. html，在网页中有个居中对齐的表格，表格的高度为 300，宽度为 200，运行效果如图 7.2 所示。

http://localhost:8080/ch7/example7_2.html

日期	天气情况
2012-05-12	晴天
2012-05-13	多云
2012-05-14	阴天

图 7.2　example7_2. html 运行效果

2. 任务的代码模板

将下列 example7_2. html 中的【代码】替换为 HTML 代码。

example7_2. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_2.html</title>
</head>
<body>
    <table【代码 1】【代码 2】【代码 3】 border=1>
    <!--代码 1 设置表格居中对齐,代码 2 设置表格的宽度为 300,代码 3 设置表格的高度为 200-->
        <tr>
            <th>日期</th>
            <th>天气情况</th>
        </tr>
        <tr>
            <td align="center">2012-05-12</td>
            <td align="center">晴天</td>
        </tr>
        <tr>
            <td align="center">2012-05-13</td>
            <td align="center">多云</td>
        </tr>
        <tr>
            <td align="center">2012-05-14</td>
            <td align="center">阴天</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

表格宽度和高度的单位既可以是像素(如300),也可以是百分比(如20%)。百分比分两种情况:如果不是嵌套表格,那么百分比是相对于浏览器窗口而言的;如果是嵌套表格,那么百分比是相对于表格所在的单元格而言的。

4. 代码模板的参考答案

【代码1】: align="center"
【代码2】: width="300"
【代码3】: height="200"

7.2.4 实践环节

把example7_2.html中的表格宽度和高度改为百分比,运行网页,并分析网页运行效果。

7.3 设置表格的边框

7.3.1 核心知识

可以使用border属性设置表格的边框,包括边框的粗细、颜色等。默认情况下,表格是没有边框的。

1. 表格边框宽度

border属性用于设置表格边框的粗细,语法格式如下:

```
<table border="边框宽度值" >
```

2. 表格边框颜色

表格边框的颜色在默认情况下是灰色的,可以使用bordercolor属性设置边框的颜色。设置边框颜色的语法格式如下:

```
<table border="边框宽度值" bordercolor="颜色值">
```

其中,边框宽度值大于0(否则无法显示边框的颜色),颜色值为十六进制的颜色值或英文的颜色名称。

3. 单元格的间距

表格内部每个单元格之间的间距可以使用cellspacing属性来设置,语法格式如下:

```
<table border="边框宽度值" cellspacing="间距值">
```

4. 单元格边框和内容之间的距离

单元格边框和内容之间的间距可以使用cellpadding属性来设置,语法格式如下:

```
<table border="边框宽度值" cellpadding="内容与边框的间距值">
```

7.3.2　能力目标

能使用 border 属性设置表格的边框，包括边框的宽度、边框的颜色、单元格的间距、单元格边框和内容的间距。

7.3.3　任务驱动

1. 任务的主要内容

编写网页 example7_3. html，在网页中有个表格，表格的边框宽度为 3，边框颜色为 green，单元格间距为 4。网页运行效果如图 7. 3 所示。

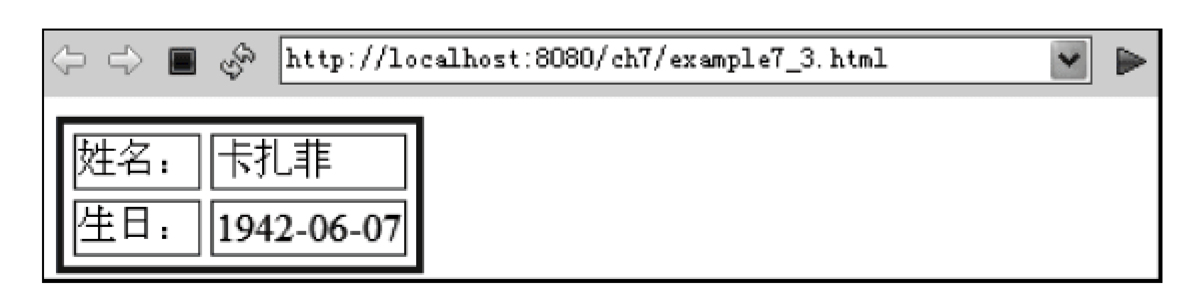


图 7. 3　example7_3. html 运行效果

2. 任务的代码模板

将下列 example7_3. html 中的【代码】替换为 HTML 代码。

example7_3. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_3.html</title>
</head>
<body>
    <table【代码 1】【代码 2】【代码 3】>
    <!--代码 1设置表格边框宽度为 3,代码 2设置边框颜色为 green,代码 3设置单元格间距为 4-->
        <tr>
            <td>姓名:</td>
            <td>卡扎菲</td>
        </tr>
        <tr>
            <td>生日:</td>
            <td>1942-06-07</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

边框宽度值一般不超过 5 个像素，过于粗大的边框会影响表格的整体美观。border 属性设置的表格边框只是表格四周的边框粗细，并不影响单元格之间的边框尺寸。

4. 代码模板的参考答案

【代码 1】: border="3"
【代码 2】: bordercolor="green"
【代码 3】: cellspacing="4"

7.3.4 实践环节

在 example7_3.html 中使用 cellpadding 属性设置单元格内容与单元格边框的间距值为 50,并比较设置前与设置后的页面运行效果,设置后的页面运行效果如图 7.4 所示。

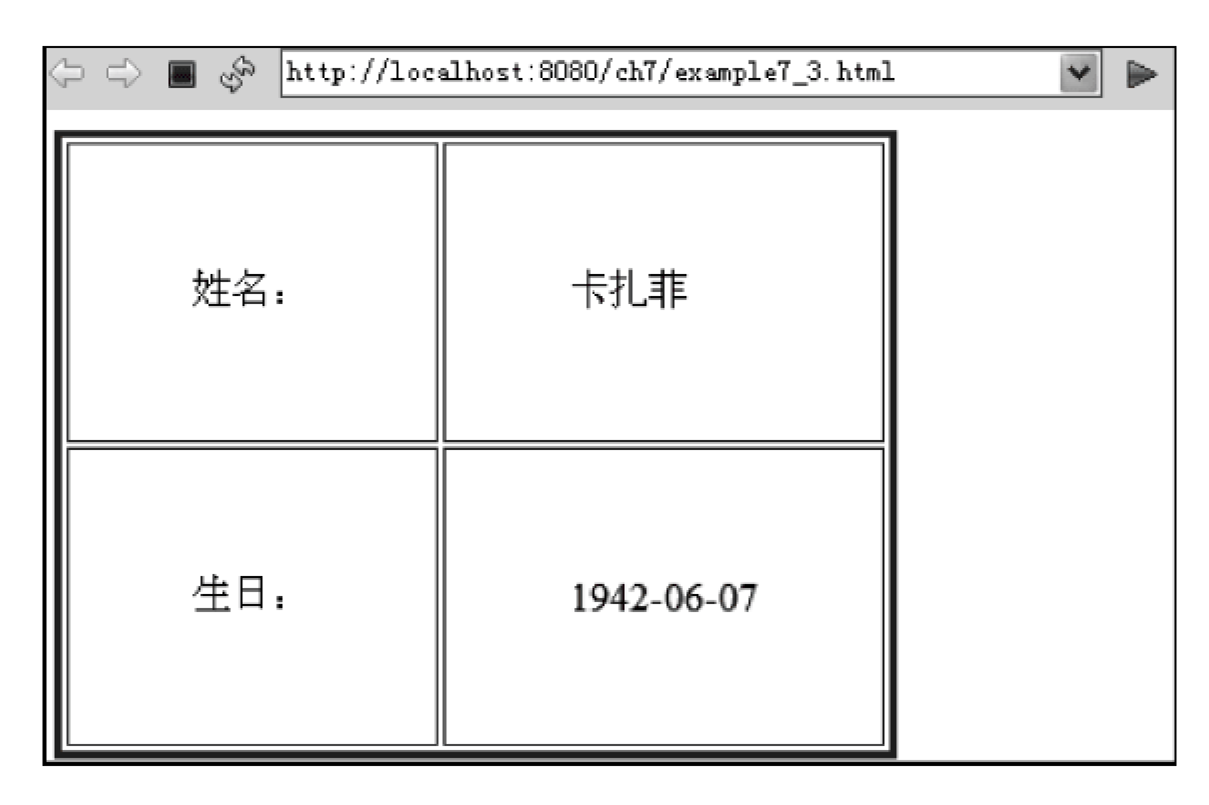


图 7.4 example7_3.html 修改后的运行效果

7.4 设置表格的背景

7.4.1 核心知识

表格在网页中创建之后,可以为表格设置不同的背景来美化表格,如设置表格的背景色和背景图像。

1. 设置表格的背景色

bgcolor 属性用来设置表格的背景颜色,语法格式如下:

```
<table bgcolor="颜色值">
```

2. 设置表格的背景图像

美化表格时,不仅可以设置表格的背景色,还可以设置表格的背景图片。设置表格背景图像的语法格式如下:

```
<table background="背景图片路径">
```

7.4.2 能力目标

能使用 bgcolor 属性设置表格的背景色,能使用 background 属性设置表格的背景图像。

7.4.3　任务驱动

1. 任务的主要内容

编写网页 example7_4. html，在网页中有个表格，表格的背景色为 lightblue。网页运行效果如图 7.5 所示。

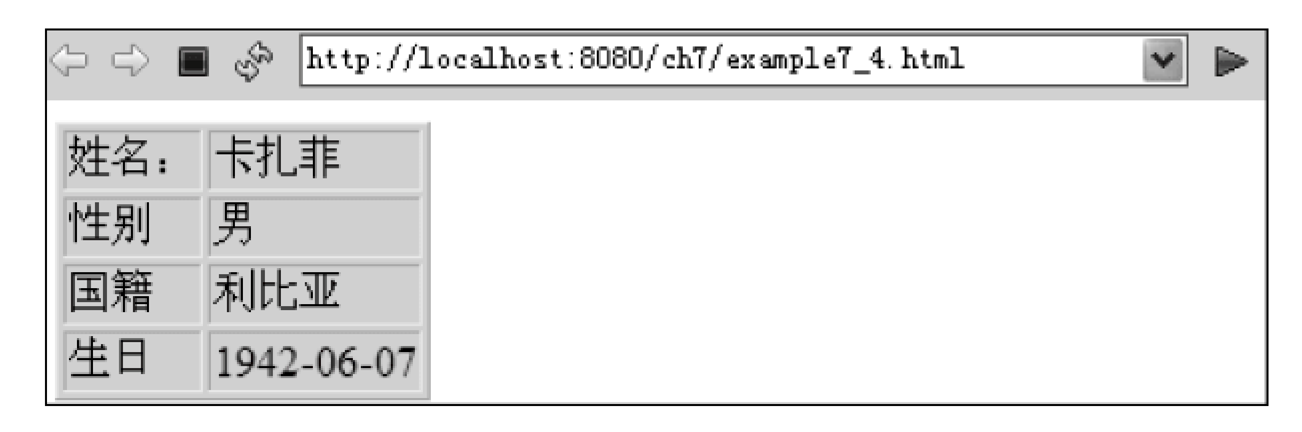


图 7.5　example7_4. html 的运行效果

2. 任务的代码模板

将下列 example7_4. html 中的【代码】替换为 HTML 代码。

example7_4. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_4.html</title>
</head>
<body>
    <table【代码 1】border="1">
    <!--代码 1 设置表格的背景色为 lightblue -->
        <tr>
            <td>姓名：</td>
            <td>卡扎菲</td>
        </tr>
        <tr>
            <td>性别</td>
            <td>男</td>
        </tr>
        <tr>
            <td>国籍</td>
            <td>利比亚</td>
        </tr>
        <tr>
            <td>生日</td>
            <td>1942-06-07</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

如果表格的行或单元格设置了背景色，那么表格的背景颜色会被覆盖。如果一个表格既设置背景色又设置背景图片，那么背景图片会覆盖掉背景色。

4. 代码模板的参考答案

【代码 1】：bgcolor="lightblue"

7.4.4 实践环节

在 example7_4.html 页面中使用 background 属性给表格设置背景图片，运行修改后的网页，并分析运行效果。

7.5 设置表格的行属性

7.5.1 核心知识

在网页中设计表格时，除了使用表格标记<table>的属性从整体上设计表格样式外，还可以使用行标记<tr>的属性逐行设计行的样式。

1. 设置行的高度

在网页中使用表格时，偶尔遇到表格中某一行的高度与其他行不同，这时就需要使用<tr>标记的 height 属性设置该行的高度，语法格式如下：

```
<tr height="行的高度值" >
```

2. 设置行文字的水平对齐方式

<tr>标记的 align 属性可以用于设置表格行内文字的水平对齐方式，语法格式如下：

```
<tr align="对齐方式" >
```

3. 设置行的边框颜色

<tr>标记的 bordercolor 属性可以用于设置行的边框颜色，语法格式如下：

```
<tr bordercolor="颜色值">
```

4. 设置行的背景颜色

<tr>标记的 bgcolor 属性可以用于设置行的背景颜色，语法格式如下：

```
<tr bgcolor="颜色值">
```

7.5.2 能力目标

能使用<tr>标记的各种属性设置行的样式，包括行的高度、边框颜色以及背景颜色。

7.5.3　任务驱动

1. 任务的主要内容

编写网页 example7_5.html，在网页中有个表格，表格共有 3 行数据，第 1 行的背景颜色为 cyan，第 3 行的边框颜色为 red。网页运行效果如图 7.6 所示。



图 7.6　example7_5.html 的运行效果

2. 任务的代码模板

将下列 example7_5.html 中的【代码】替换为 HTML 代码。

example7_5.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_5.html</title>
</head>
<body>
    <table border="1">
        <tr【代码 1】><!--代码 1 设置行的背景色为 cyan -->
            <th>姓名：</th>
            <td>葫芦娃</td>
            <th>性别：</th>
            <td>男</td>
        </tr>
        <tr>
            <th>国籍：</th>
            <td>中国</td>
            <th>生日：</th>
            <td>2002-06-07</td>
        </tr>
        <tr【代码 2】><!--代码 2 设置行的边框颜色为 red -->
            <th>政治面貌：</th>
            <td>群众</td>
            <th>爱好：</th>
            <td>睡觉</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

设置表格的行属性是针对某行有效的,如果某行的样式与表格的样式相冲突,那么在该行中,行的样式将覆盖表格的样式。

4. 代码模板的参考答案

【代码 1】: bgcolor="cyan"
【代码 2】: bordercolor="red"

7.5.4 实践环节

编写网页 practice7_5.html,该网页中有个表格,第一行的高度为 50,第一行的背景色为 lightgreen,第一行的文字内容右对齐。页面运行效果如图 7.7 所示。

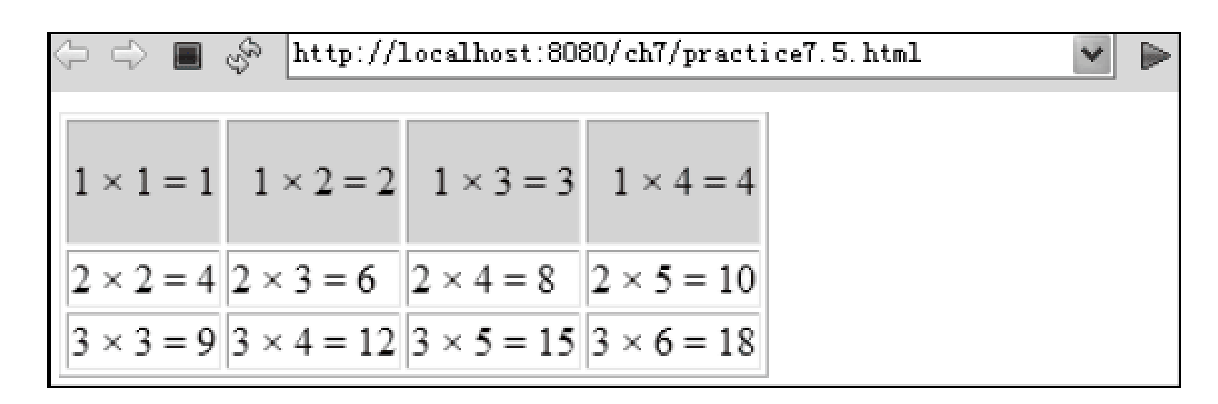


图 7.7 practice7_5.html 的运行效果

7.6 设置表格的单元格属性

一个表格可以有多行,一行可以有多个单元格。单元格是表格中最基本的单位。<td>单元格标记都要包含在行标记<tr>里面,一行可以有任意多个单元格。程序员可以设置单元格的各种样式,这些样式将覆盖<table>和<tr>已经定义好的样式。

7.6.1 核心知识

1. 设置单元格的宽度和高度

在默认状态下,单元格的大小会根据自己的内容自动调整,程序员可以通过 width 和 height 调整单元格的宽度和高度,语法格式如下:

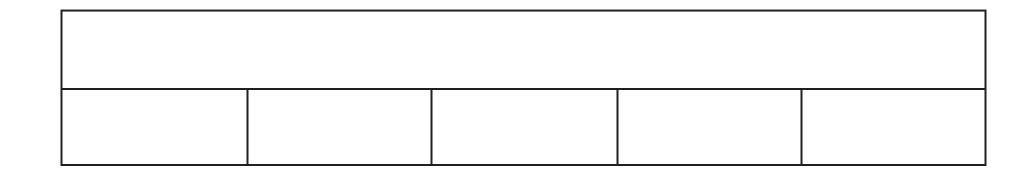
```
<td width="单元格的宽度值" height="单元格的高度值">
```

2. 设置单元格的水平跨度(跨多列)

在制作表格时,可能需要某个单元格占据多列的位置,这时候就要使用 colspan 属性设置该单元格所跨的列数,语法格式如下:

```
<td colspan="跨的列数值">
```

如下表第一行的单元格的水平跨度为 5。

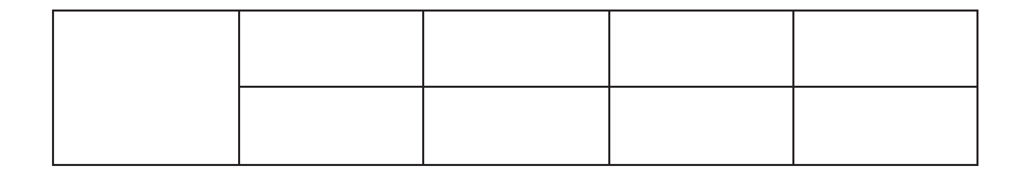


3. 设置单元格的垂直跨度(跨多行)

在制作表格时,可能需要某个单元格占据多行的位置,这时候就要使用 rowspan 属性设置该单元格所跨的行数。

```
<td rowspan="跨的行数值">
```

如下表第一个单元格的垂直跨度为 2。



4. 设置单元格的对齐方式

单元格对齐方式的设置方法与行对齐方式的设置方法一样,读者可以参考＜tr＞标记的 align 和 valign 属性来设置单元格的对齐方式。

5. 设置单元格的背景色、边框颜色

单元格的背景色、边框颜色的设置方法与行的背景色、边框颜色的设置方法一样,读者可以参考＜tr＞标记的 bgcolor 和 bordercolor 属性来设置单元格的背景色、边框颜色。

7.6.2　能力目标

能使用＜td＞标记的各种属性设置单元格的样式,包括高度、宽度、水平跨度、垂直跨度、边框颜色以及背景颜色。

7.6.3　任务驱动

1. 任务的主要内容

编写网页 example7_6. html,在网页中有个表格,表格标题为"个人简历",边框宽度为 1,边框颜色为 green。网页运行效果如图 7.8 所示。



图 7.8　example7_6. html 的运行效果

2. 任务的代码模板

将下列 example7_6. html 中的【代码】替换为 HTML 代码。

example7_6. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_6.html</title>
</head>
<body>
    <table border="1" bordercolor="green">
        <caption>个人简历</caption>
        <tr>
            <th【代码 1】 align="left">基本资料</th><!--代码 1 设置单元格的垂直跨度为 2
             -->
            <th align="left">姓名</th>
            <td>孙悟空</td>
            <th align="left">性别</th>
            <td>男</td>
        </tr>
        <tr>
            <th align="left">政治面貌</th>
            <td>群众</td>
            <th align="left">出生日期</th>
            <td>1988-12-09</td>
        </tr>
        <tr>
            <th【代码 2】>业余爱好</th><!--代码 2 设置单元格的水平跨度为 5 -->
        </tr>
        <tr>
            <td>替身</td>
            <td>各种球类(包括中国足球)</td>
            <td>看言情</td>
            <td>爬山</td>
            <td>压马路</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

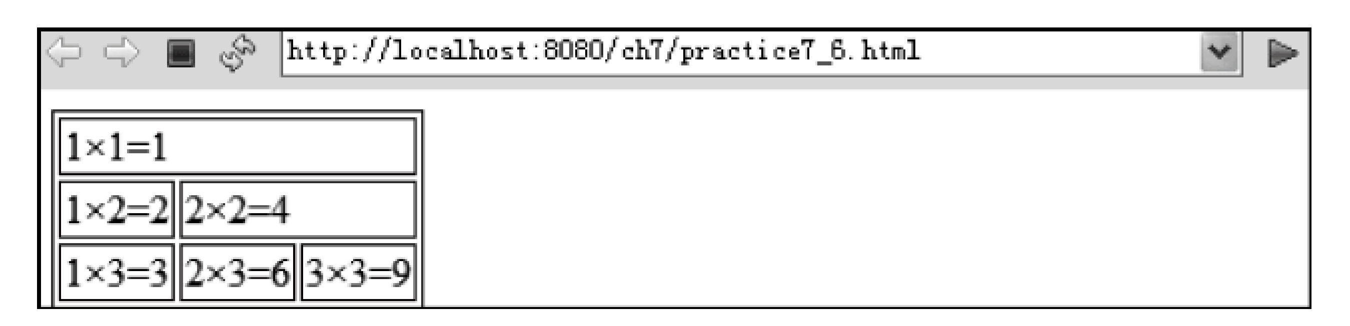
设置表格的单元格属性是针对某个单元格有效的,如果某个单元格的样式与表格或行的样式相冲突,那么在该单元格中,单元格的样式将覆盖表格或行的样式。

4. 代码模板的参考答案

【代码 1】: rowspan="2"
【代码 2】: colspan="5"

7.6.4 实践环节

编写网页 practice7_6.html,在网页中有个表格,边框宽度为 1,边框颜色为 blue。网页运行效果如图 7.9 所示。



1×1=1		
1×2=2	2×2=4	
1×3=3	2×3=6	3×3=9

图 7.9　practice7_6.html 的运行效果

7.7　表格的嵌套

7.7.1　核心知识

设计表格时，可能需要在表格的某个单元格里面放置很复杂的内容，这就需要在单元里面再嵌套一个表格，称为表格的嵌套。表格嵌套的语法格式如下：

```
<table>
    <tr>
        <td>单元格中的内容</td>
        <td>单元格中的内容</td>
        …
    </tr>
    <tr>
        <td>
            <table>…</table>
        </td>
        <td>单元格中的内容</td>
        …
    </tr>
    …
</table>
```

其中，任何一对<td>和</td>标记里面都可以嵌套一个表格，也就是说表格一定要嵌套在单元格里面。

7.7.2　能力目标

能灵活使用表格的嵌套原理布局复杂的页面内容。

7.7.3　任务驱动

1. 任务的主要内容

编写网页 example7_7.html，在网页中有个表格，该表格共有两行，第 1 行有两个单元格，在第 1 个单元格里面嵌套 1 个 3 行 1 列的表格，在第 2 个单元格里面嵌套 1 个 1 行 3 列的表格。网页运行效果如图 7.10 所示。

2. 任务的代码模板

将下列 example7_7.html 中的【代码】替换为 HTML 代码。

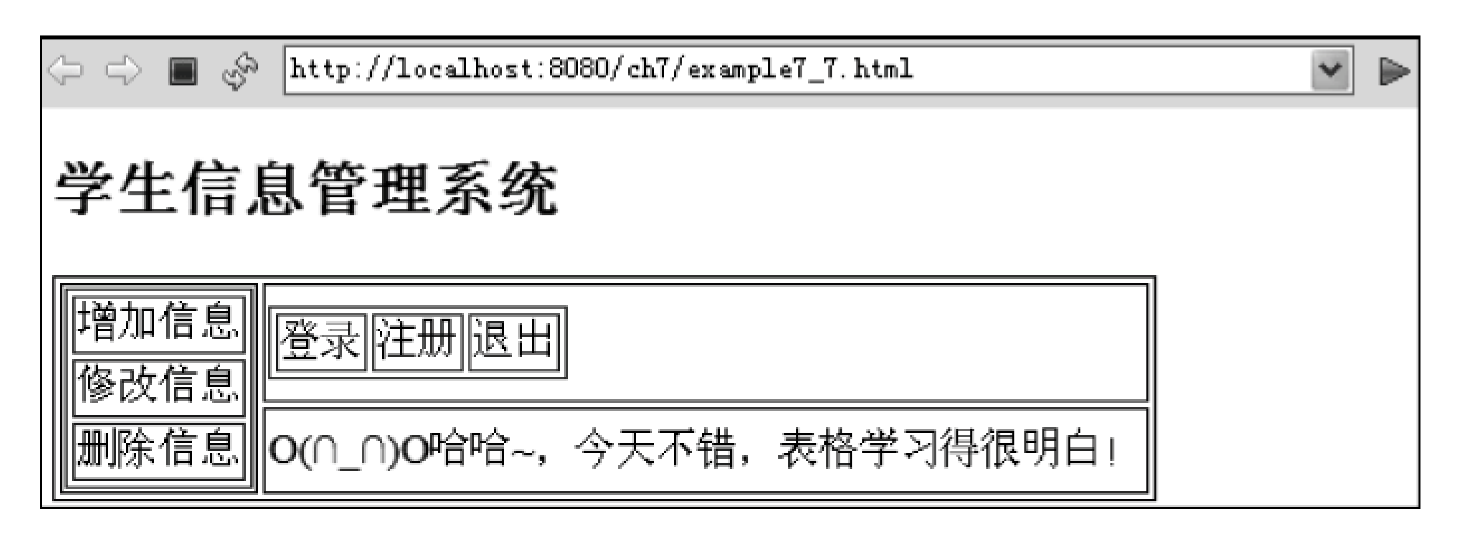


图 7.10　example7_7.html 的运行效果

example7_7.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_7.html</title>
</head>
<body>
    <h2>学生信息管理系统</h2>
    <table border="1" bordercolor="blue">
        <tr>
            <td rowspan="2">
                【代码 1】
                <!--代码 1 嵌套一个 3 行 1 列的表格 -->
            </td>
            <td>
                【代码 2】
                <!--代码 2 嵌套一个 1 行 3 列的表格 -->
            </td>
        </tr>
        <tr>
            <td>o(∩_∩)o 哈哈~ ,今天不错,表格学习得很明白!</td>
        </tr>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

在 HTML 页面中,使用表格排版是通过嵌套来完成的,即一个表格内部可以嵌套另一个表格。用表格来排版页面的思路是:由总表格规划整体的结构,由嵌套的表格负责各个子栏目的排版,并插入到表格的相应位置,这样就可以使页面的各个部分有条不紊,互不冲突,看上去清晰整洁。在实际做网页时一般不显示边框,边框的显示可根据自己的爱好来设定。在实例中为了让读者能够看清楚,都设置了边框。

4. 代码模板的参考答案

【代码 1】: `<table border="1" bordercolor="red">`

```
        <tr>
            <td>增加信息</td>
        </tr>
        <tr>
            <td>修改信息</td>
        </tr>
        <tr>
            <td>删除信息</td>
        </tr>
    </table>
```

【代码 2】:
```
    <table border="1" bordercolor="red">
        <tr>
            <td>登录</td>
            <td>注册</td>
            <td>退出</td>
        </tr>
    </table>
```

7.7.4　实践环节

编写网页 practice7_7.html，网页中有个表格，在该表格第 2 行的每一个单元格中分别嵌套了一个表格。网页运行效果如图 7.11 所示。



图 7.11　practice7_7.html 的运行效果

7.8　表格的结构

7.8.1　核心知识

为了在设计网页时，能清楚地区分表格的结构，HTML 语言中有<thead>、<tbody>

和<tfoot>3 个标记分别表示一个表格的表头、表主体和表尾。

1. 表格的表头标记<thead>

表头标记用于定义表格最顶端表头的样式，包括背景颜色、背景图片、文字对齐方式等属性，具体语法格式如下：

```
<table>
    <thead bgcolor="背景颜色" align="水平对齐方式">
        <tr>
            <td>表头单元格内容</td>
            <td>表头单元格内容</td>
            …
        </tr>
    </thead>
    …
</table>
```

2. 表格的表主体标记<tbody>

表主体标记<tbody>与表头标记的功能类似，用来统一设置表主体部分的样式，具体语法格式如下：

```
<table>
    <thead bgcolor="red" align="center">
        <tr>
            <td>表头单元格内容</td>
            <td>表头单元格内容</td>
            …
        </tr>
    </thead>
    <tbody bgcolor="背景颜色" align="水平对齐方式">
        <tr>
            <td>表主体单元格内容</td>
            <td>表主体单元格内容</td>
            …
        </tr>
        …
    </tbody>
    …
</table>
```

3. 表格的表尾标记<tfoot>

表尾标记<tfoot>与表头标记<thead>和表主体标记<tbody>一样，用来统一设置表尾部分的样式，具体语法格式如下：

```
<table>
    <thead bgcolor="red" align="center">
        <tr>
            <td>表头单元格内容</td>
            <td>表头单元格内容</td>
```

```
            …
        </tr>
    </thead>
    <tbody bgcolor="green" align="left">
        <tr>
            <td>表主体单元格内容</td>
            <td>表主体单元格内容</td>
            …
        </tr>
        …
    </tbody>
    <tfoot bgcolor="背景颜色" align="水平对齐方式">
        <tr>
            <td>表尾单元格内容</td>
            <td>表尾单元格内容</td>
            …
        </tr>
        …
    </tfoot>
</table>
```

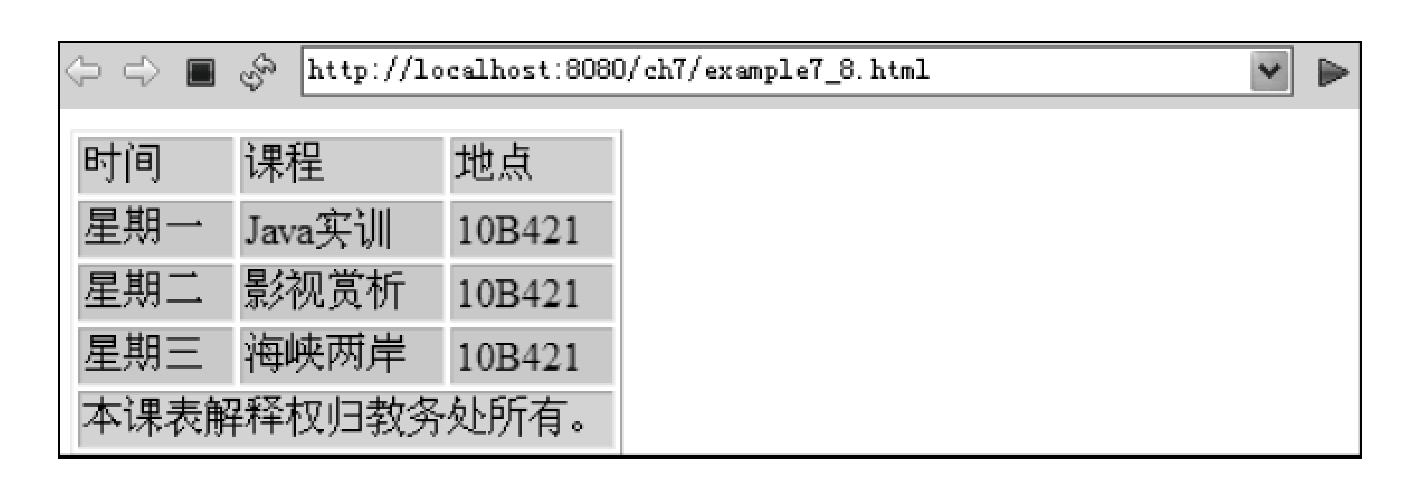
7.8.2　能力目标

能使用<thead>、<tbody>和<tfoot>标记创建表格的表头、表主体和表尾部分。

7.8.3　任务驱动

1. 任务的主要内容

编写网页 example7_8. html，在该网页中有个表格，表格包括表头、表主体、表尾 3 部分。表格的第 1 行为表头，背景色为“LightBLue”。表格的第 2～4 行为表主体，背景色为“Cyan”。表格的第 5 行为表尾，背景色为“LightGreen”。网页运行效果如图 7. 12 所示。



时间	课程	地点
星期一	Java实训	10B421
星期二	影视赏析	10B421
星期三	海峡两岸	10B421
本课表解释权归教务处所有。		

图 7. 12　example7_8. html 的运行效果

2. 任务的代码模板

将下列 example7_8. html 中的【代码】替换为 HTML 代码。

example7_8. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example7_8.html</title>
</head>
<body>
    <table border="1">
        【代码 1】<!--代码 1 创建表头,表头背景色为 LightBLue -->
            <tr>
                <td>时间</td>
                <td>课程</td>
                <td>地点</td>
            </tr>
        </thead>
        【代码 2】<!--代码 2 创建表主体,表主体背景色为 Cyan -->
            <tr>
                <td>星期一</td>
                <td>Java 实训</td>
                <td>10B421</td>
            </tr>
            <tr>
                <td>星期二</td>
                <td>影视赏析</td>
                <td>10B421</td>
            </tr>
            <tr>
                <td>星期三</td>
                <td>海峡两岸</td>
                <td>10B421</td>
            </tr>
        </tbody>
        【代码 3】<!--代码 3 创建表尾,表尾背景色为 LightGreen -->
            <tr>
                <td colspan="3">本课表解释权归教务处所有。</td>
            </tr>
        </tfoot>
    </table>
</body>
</html>
```

3. 任务小结或知识扩展

在<thead>、<tbody>和<tfoot>标记中可以包含<tr>、<th>和<td>标记。在实际的网页制作中,表头、表主体和表尾部分的样式经常由 CSS 指定。

4. 代码模板的参考答案

```
【代码 1】:<thead bgcolor="LightBLue">
【代码 2】:<tbody bgcolor="Cyan">
【代码 3】:<tfoot bgcolor="LightGreen">
```

7.8.4　实践环节

在网页 example7_8.html 中，如果不使用表头、表主体和表尾标记，如何实现该网页？并比较哪种实现方式更简便。

7.9　小　　结

标记 table 创建表格，表格里面又有行和列。标记 tr 创建行，标记 td 创建单元格。行是水平的，贯穿表格的左右，列则是垂直的，单元格是行和列交汇的部分，是输入和显示信息的地方。

通过 table 标记的属性可以给整个表格设置样式，通过 tr 标记的属性可以给表格的某一行设置样式，通过 td 标记的属性可以给某个单元格设置样式。但要注意，行的样式会覆盖表格的样式，单元格的样式会覆盖行的样式。

使用 colspan 属性可以设置该单元格所跨的列数（水平跨度），使用 rowspan 属性可以设置该单元格所跨的行数（垂直跨度）。

习　题　7

1. 要使表格的边框不显示，应设置 border 的值是（　　）。
   A. 1　　B. 0　　C. 2　　D. 3
2. 以下标记中，用于定义一个单元格的是（　　）。
   A. <td> </td>
   B. <tr>…</tr>
   C. <table>…</table>
   D. <caption>…</caption>
3. 用于设置表格背景颜色的属性的是（　　）。
   A. background　　B. bgcolor
   C. bordercolor　　D. backgroundcolor
4. 表格的开始标记为（　　）。
   A. <table>　　B. </table>　　C. <p>　　D. <tr>
5. 以下属性可以添加在<table>标记后面的是（　　）。
   A. border　　B. width　　C. height　　D. 以上都可以
6. 下列对于<th>标记和<td>标记说法不正确的是（　　）。
   A. <th>和<td>都可以标记一个单元格
   B. 有多少个单元格就有多少个<th>或<td>
   C. <th>标记所标记的单元格的文字以粗体出现
   D. <th>和<td>标记作用完全一样
7. 下列对于表格的说法正确的是（　　）。
   A. 不可以使用表格对页面中的内容进行排版

B. rowspan属性用于定义有横向通栏的表格

C. colspan属性用于定义有纵向通栏的表格

D. 在页面布局时为了页面的美观也经常用到表格，但我们却看不见这些表格，是由于border属性的值被设定为“0”了

8. 下面对于bgcolor属性说法正确的是(　　)。

A. 可以用来定义表格的颜色

B. <table>标记中必须出现bgcolor属性

C. bgcolor属性的值必须是十六进制的6位数

D. bgcolor属性的值必须是已经定义好的颜色

第 8 章

表单与页面控件

主要内容

- 表单标记
- 表单对象
- 下拉列表
- 文本区标记

在制作动态网页时，常用表单收集用户的信息，将用户的信息通过表单封装提交给服务器，从而实现与用户的交互。因此，表单是 HTML 页面与服务器端实现交互的重要手段。本章将重点介绍表单标记以及各种表单对象的用法。

8.1 表单标记 form

在 HTML 网页中，表单是用＜form＞标记定义的，它类似于一个容器，表单对象必须在表单中才有效，如 input。

8.1.1 核心知识

1. 处理动作 action

＜form＞标记里的 action 属性用于指定表单数据提交给哪个程序进行处理，语法格式如下：

```
<form action="表单的处理程序">
```

其中，表单的处理程序是表单标记必不可少的参数，它是表单要提交的地址，换句话说是表单收集到的信息要传递到的程序地址，如某页面。

2. 表单名称 name

在一个网页程序中，可能需要多个表单来提交信息，这时就需要用 name 属性给表单命一个名称。表单名称可以控制表单与后台程序之间的关系，语法格式如下：

```
<form action="表单的处理程序" name="表单的名称">
```

其中，表单的名称是一个标识符，不能含有特殊的字符和空格。

3. 表单提交方法 method

<form>标记里的 method 属性用于指定使用哪种提交方法将表单数据提交到服务器。默认情况下，提交方法为 get，但 get 方法不具有保密性，而且不能提交非 ASCII 的字符。语法格式如下：

```
<form action="表单的处理程序" method="提交方法" name="表单的名称">
```

其中，表单的提交方法有两种：get 和 post。get 方法将表单内容以附件形式放在 URL 地址后面，因此有长度限制（最大 8192 个字符），而且不安全。post 方法将用户在表单中输入的数据包含在表单主体中，一起提交给服务器，该方法没有信息长度的限制，也比较安全。

4. 编码方式 enctype

<form>标记里的 enctype 属性用于指定表单信息提交的编码方式，这个编码方式通常情况下采用默认的（application/x-www-form-urlencoded）即可，但上传文件时必须选择 mime 编码（multipart/form-data）。表单编码方式的语法格式如下：

```
<form action="表单的处理程序" method="提交方法" enctype="编码方式" name="表单的名称">
```

其中，编码方式有 3 种情况：text/plain（纯文本形式）、application/x-www-form-urlencoded 和 multipart/form-data。

8.1.2 能力目标

理解并掌握表单标记的 action 属性、method 属性和 enctype 属性的用法。

8.1.3 任务驱动

1. 任务的主要内容

编写网页 example8_1. html，具体要求如下：

- 网页中有个 h3 标题"认真学习表单标记<form>"；
- 网页中有个表单，表单的处理程序是 action. html，表单的提交方法是 post，编码方式采用默认的；
- 网页 example8_1. html 的运行效果如图 8.1 所示。

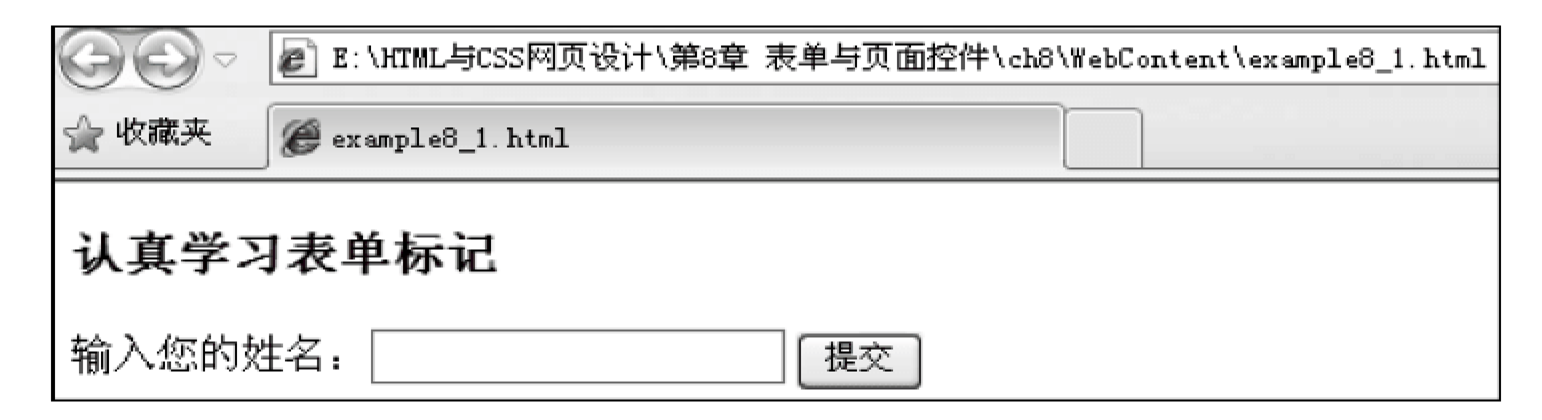


图 8.1　example8_1. html 运行效果

2. 任务的代码模板

将下列 example8_1. html 中的【代码】替换为 HTML 代码。

example8_1. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_1.html</title>
</head>
<body>
    <!--代码 1 设置表单的处理程序为 action.html,代码 2 设置表单的提交方式为 post -->
    <form【代码 1】【代码 2】>
        输入您的姓名：<input type="text" name="yourName"/>
        <input type="submit" value="提交"/>
    </form>
</body>
</html>
```

action. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>action.html</title>
</head>
<body>
    来自 example8_1.html 页面的请求！
</body>
</html>
```

3. 任务小结或知识扩展

表单的处理程序由 action 属性值指定，如果该属性值为空，表单的处理程序就是表单所在的页面本身。

4. 代码模板的参考答案

```
【代码 1】: action="action.html"
【代码 2】: method="post"
```

8.1.4　实践环节

把 example8_1. html 页面中表单的提交方法改成 get，用 IE 浏览器运行 example8_1. html 页面，输入姓名后单击“提交”按钮，仔细观察浏览器地址栏中 URL 的变化。

8. 2　input 标记——文本框与密码框

在表单中，input 标记是最常用的表单标记。input 标记里面有个 type 属性值确定表单对象的类型，如 text 代表的是文本框类型。另外，input 标记必须放在<form>与</form>标

记之间才有效。

8.2.1 核心知识

1. 文本框

如果 input 标记的 type 属性值为 text，那么这个 input 标记(插入表单对象)代表的就是单行文本框，可以输入任何类型的文本、数字或字母，输入的内容是单行显示。文本框的语法格式如下：

```
<input type="text" value="文本框内的默认值" name="文本框的名称" size="文本框的长度"
maxlength="最大字符数"/>
```

其中，type 属性指定插入哪种表单对象，如 type＝"text"，即插入的是文本框(文字内容)；value 属性设置文本框的默认值；name 属性是文本框的名字，用于和页面中的其他控件进行区分，处理程序通过 name 获取 value 的值；size 属性值确定文本字段在页面中显示的长度，以字符为单位；maxlength 属性值用于设定文本框中最大可以输入的字符数，如下面的代码。

```
姓名:<input type="text" value="" name="userName"/>
```

上述代码呈现的结果如图 8.2 所示。

2. 密码框

如果 input 标记的 type 属性值为 password，那么这个 input 标记代表的就是密码框，在其中输入的字符都是以"＊"或圆点"•"显示。密码框的语法格式如下：

```
<input type="password" value="密码框的默认值" name="密码框的名称" size="密码框的长度"
maxlength="最大字符数"/>
```

其中的属性和文本框中的属性类似，如下面的代码。

```
密码:<input type="password" value="123456" name="pwd"/>
```

上述代码呈现的结果如图 8.3 所示。

姓名：[ ]

图 8.2　文本框呈现结果

密码：●●●●●●

图 8.3　密码框呈现结果

8.2.2 能力目标

能使用 input 设计文本框和密码框。

8.2.3 任务驱动

1. 任务的主要内容

编写网页 example8_2.html，具体要求如下：

• 网页中有个表单，表单的处理程序是页面本身；

- 在表单中有个文本框，文本框的最大输入字符数为 6；
- 在表单中还有个密码框，网页运行效果如图 8.4 所示。

图 8.4　example8_2. html 页面运行效果

2. 任务的代码模板

将下列 example8_2. html 中的【代码】替换为 HTML 代码。

example8_2. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_2.html</title>
</head>
<body>
    <form action="" method="post">
        <!--代码 1 设置 type 的值为 text,代码 2 设置文本框最大输入字符数为 6 -->
        账号:<input【代码 1】name="ID" value="000001"【代码 2】/><br>
        <!--代码 2 设置 type 的值为 password -->
        密码:<input【代码 3】name="PWD" value="123456"/>
    </form>
</body>
</html>
```

3. 任务小结或知识扩展

从 example8_2. html 页面的运行效果看，文本框与密码框的默认长度是不一样的。一般情况下，为了页面美观需要使用 CSS 把文本框与密码框的长度设置成相同。

在文本框与密码框中都有 size 属性，其最小值为 1，最大值由浏览器的宽度决定。

4. 代码模板的参考答案

```
【代码 1】: type="text"
【代码 2】: maxlength="6"
【代码 3】: type="password"
```

8.2.4　实践环节

使用 size 属性把 example8_2. html 中文本框的文本字段在页面中显示的长度设置为 5，然后运行 example8_2. html 页面，仔细观察运行结果。

8.3 input 标记——单选按钮与复选框

8.3.1 核心知识

1. 单选按钮

单选按钮用来让用户进行单一选择，如让用户选择他的性别。单选按钮在页面中以圆框（“○”）显示，语法格式如下：

```
<input type="radio" value="单选按钮的值" name="单选按钮的名称" checked/>
```

其中，当用户选中单选按钮后，value 属性的值会传递给处理程序。name 代表单选按钮的名称，一组单选按钮的名称都相同，处理程序通过 name 获取被选中的单选按钮的 value 值。checked 表示该单选按钮被选中，而在一组单选按钮中只有一个单选按钮设置为 checked。如下面的代码：

```
<input type="radio" value="male" name="sex" checked/>男
<input type="radio" value="female" name="sex" />女
```

上述代码呈现的结果如图 8.5 所示。

2. 复选框

我们经常看到这样的问题，“请选择您喜欢的歌手：□顾小白 □左永邦 □罗书权”，这样的网页就是使用复选框实现的。复选框与单选按钮的不同之处是复选框能够实现选项的多选，以一个方框（“□”）表示，语法格式如下：

```
<input type="checkbox" value="复选框的值" name="复选框的名称" checked/>
```

其中，当用户选中复选框后，value 属性的值传递给处理程序。name 代表的是复选框的名称，一组复选框的名称都相同，处理程序通过 name 获取被选中的复选框的 value 值（以数组的形式返回，数组元素为被选中的复选框的 value 值）。checked 表示该复选框被选中，一组复选框中可以同时有多个被选中。如下面的代码：

```
<input type="checkbox" value="sunwukong" name="lover" checked/>孙悟空
<input type="checkbox" value="zhubajie" name="lover" checked/>猪八戒
<input type="checkbox" value="shaseng" name="lover" />沙僧
```

上述代码呈现的结果如图 8.6 所示。

◉男 ○女

图 8.5 单选按钮呈现结果

☑孙悟空 ☑猪八戒 □沙僧

图 8.6 复选框呈现结果

8.3.2 能力目标

理解并掌握单选按钮与复选框的使用方法。

8.3.3　任务驱动

1. 任务的主要内容

编写网页 example8_3. html,具体要求如下:

- 网页中有个 h3 标题“调查表”;
- 网页中有个表单,表单的处理程序是页面本身;
- 网页中有个文本框用来输入用户的姓名;
- 网页中有个密码框用来输入用户的密码;
- 网页中有个单选按钮供用户选择自己的性别;
- 网页中有个复选框供用户选择自己喜欢的城市;
- 网页运行效果如图 8.7 所示。

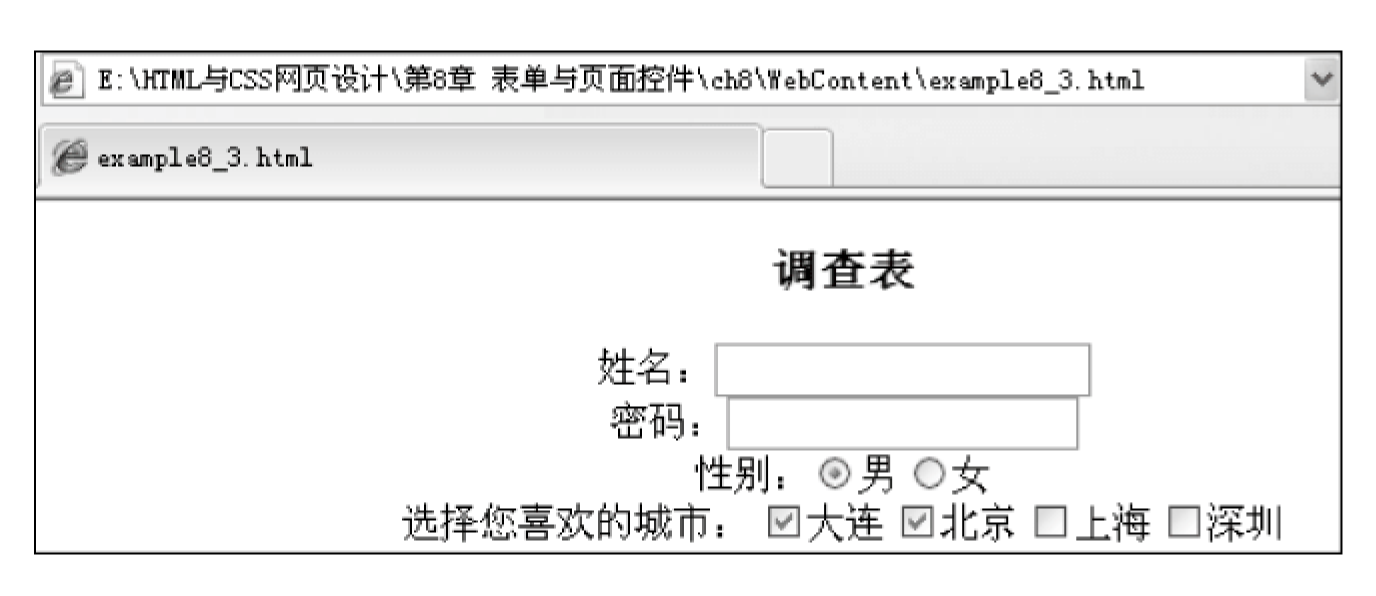


图 8.7　example8_3. html 页面运行效果

2. 任务的代码模板

将下列 example8_3. html 中的【代码】替换为 HTML 代码。

example8_3. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_3.html</title>
</head>
<body>
    <center>
    <h3>调查表</h3>
    <form action="" method="post">
        姓名:<input type="text" name="userName"/><br>
        密码:<input type="password" name="pwd"/><br>
        性别:<input【代码 1】value="male" name="sex" checked/>男
              <input type="radio" value="female"【代码 2】/>女
        <br>
        选择您喜欢的城市:
        <input【代码 3】value="DaLian" name="lover" checked/>大连
        <input type="checkbox" value="BeiJing" name="lover" checked/>北京
```

```
        <input type="checkbox" value="ShangHai" name="lover" />上海
        <input type="checkbox" value="ShenZhen" name="lover" />深圳
    </form>
    </center>
</body>
</html>
```

3. 任务小结或知识扩展

单选按钮用来表示一系列的互斥选项，这些互斥项常常被分成若干组，每组仅允许用户选择一个选项；复选框与单选按钮类似，不同之处是复选框代表多重选择，用户可以选择一个或多个选项。

4. 代码模板的参考答案

【代码 1】：type="radio"
【代码 2】：name="sex"
【代码 3】：type="checkbox"

8.3.4 实践环节

编写一个网页 practice8_3.html，页面运行效果如图 8.8 所示。

图 8.8 practice8_3.html 页面运行效果

8.4 input 标记——按钮

在网页的表单中，按钮起到至关重要的作用。如果没有按钮，那么网页很难和用户进行互动。单击按钮可以激发提交表单的动作（“提交”按钮），也可以将表单恢复到初始的状态（“重置”按钮），还可以根据程序的要求，发挥其他的作用（普通按钮）。

8.4.1 核心知识

1. 普通按钮

普通按钮主要是配合脚本语言（JavaScript）来进行表单的处理，语法格式如下：

```
<input type="button" value="按钮的值" name="按钮的名称"/>
```

其中，value 的取值就是显示在按钮上的文字，在普通按钮中可以添加 onclick、onfocus 等

JavaScript 事件实现特定的功能。

2. “重置”按钮

当用户在表单中输入信息后，想清除输入的信息，将表单恢复成初始状态时，需要使用“重置”按钮，语法格式如下：

```
<input type="reset" value="按钮的值" name="按钮的名称"/>
```

其中，value 的取值就是显示在按钮上的文字。

3. “提交”按钮

用户单击“提交”按钮时，可以实现表单内容的提交，语法如下：

```
<input type="submit" value="按钮的值" name="按钮的名称"/>
```

其中，value 的取值就是显示在按钮上的文字。

8.4.2 能力目标

能使用 input 标记创建普通按钮、“重置”按钮以及“提交”按钮，并且理解普通按钮、“重置”按钮以及“提交”按钮的使用方法。

8.4.3 任务驱动

1. 任务的主要内容

编写网页 example8_4. html，具体要求如下：

- 网页中有 3 个按钮，类型分别是“提交”按钮、“重置”按钮与普通按钮；
- 页面运行效果如图 8. 9 所示。

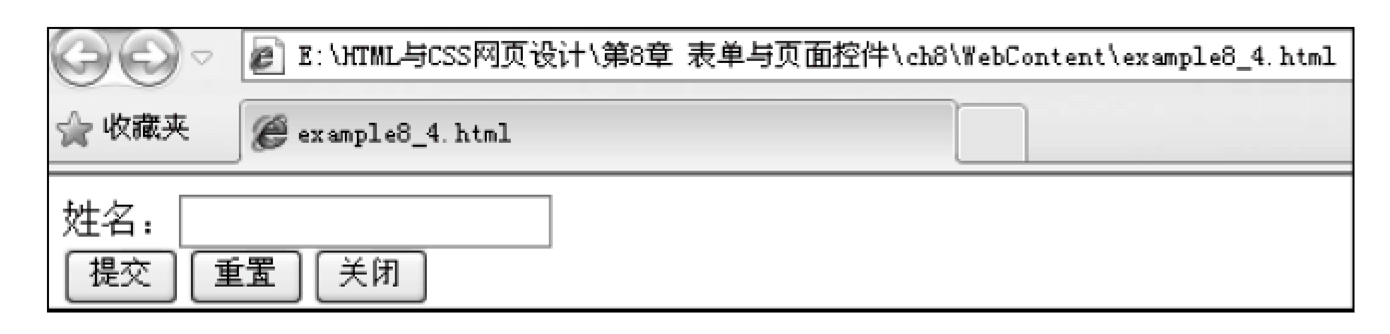


图 8. 9　example8_4. html 页面运行效果

2. 任务的代码模板

将下列 example8_4. html 中的【代码】替换为 HTML 代码。

example8_4. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_4.html</title>
</head>
<body>
    <form action="">
```

```
            姓名：<input type="text" name="userName"/><br>
            <!--代码 1 创建"提交"按钮 -->
            <input【代码 1】value="提交"/>
            <!--代码 2 创建"重置"按钮 -->
            <input【代码 2】value="重置"/>
            <!--代码 3 创建普通按钮 -->
            <input【代码 3】value="关闭" onclick="window.close()"/>
        </form>
    </body>
</html>
```

3. 任务小结或知识扩展

在 example8_4.html 页面中输入姓名后，单击“提交”按钮时，把页面的表单信息提交给表单处理程序，这时 IE 浏览器的地址栏会发生变化；单击“重置”按钮时，输入的姓名被清空，这时 IE 浏览器的地址栏没有任何变化；单击“关闭”按钮时，输入的姓名不被清空，这时 IE 浏览器的地址栏也没有任何变化，但通过 onclick 事件弹出一个关闭窗口提示框。

4. 代码模板的参考答案

```
【代码 1】: type="submit"
【代码 2】: type="reset"
【代码 3】: type="button"
```

8.4.4 实践环节

编写一个网页 practice8_4.html，具体要求如下：

- 网页中表单的处理程序是页面本身；
- 表单中有文本框、密码框、单选按钮、复选框以及 3 种按钮（“提交”按钮、“重置”按钮与普通按钮）；
- 页面运行效果如图 8.10 所示。

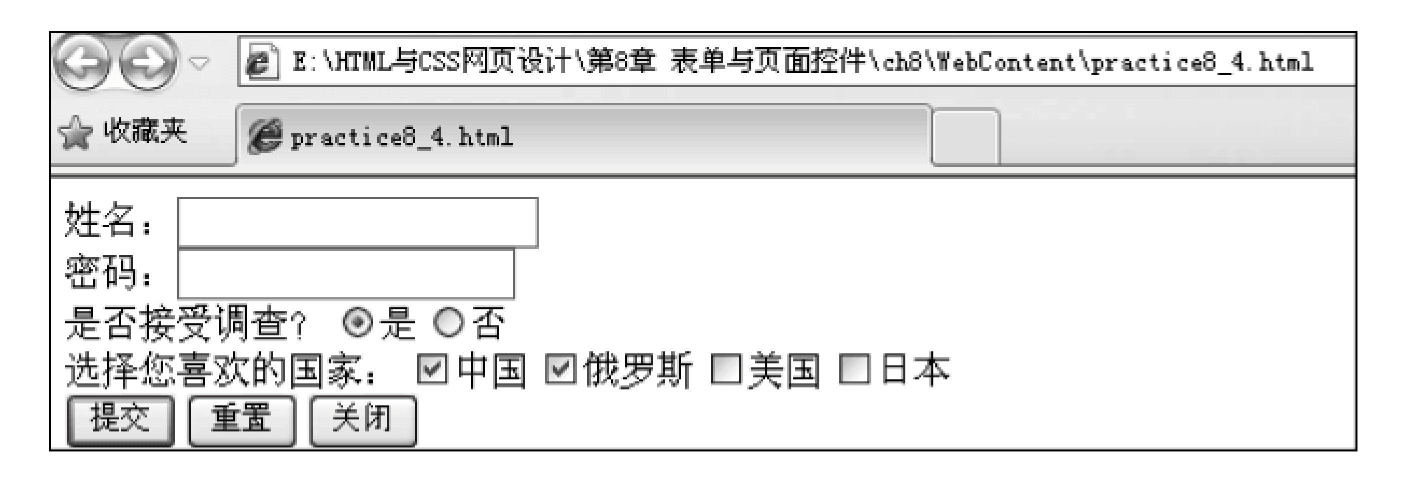


图 8.10 practice8_4.html 页面运行效果

8.5 input 标记——图像域、隐藏域以及文件域

8.5.1 核心知识

1. 图像域

我们在制作网页时，有时发现按钮的外观很单调，甚至感觉按钮破坏了整体的设计。这

时，可以使用图像域创建和网页整体效果一致的图像“提交”按钮。因此，图像域是指用在“提交”按钮位置的图像，使得该图像具有“提交”按钮的功能。

```
<input type="image" name="图像域的名称" src="图像的路径"/>
```

其中，图像的路径可以是绝对路径，也可以是相对路径。

2. 隐藏域

隐藏域在页面中用户是看不到的。使用隐藏域的目的主要是收集和发送信息，以便于被表单的处理程序所使用。提交表单时，隐藏域的值一块被发送给服务器端。在表单中可以根据需要使用任意个隐藏域，语法格式如下：

```
<input type="hidden" name="隐藏域的名称" value="隐藏域的值"/>
```

3. 文件域

我们平时在网站上经常用到上传照片或文件的功能。上传文件的控件就是文件域，它是由一个文本框和一个“浏览”按钮组成的。用户上传文件时，可以直接在文本框中输入要上传文件的路径，也可以单击“浏览”按钮选择文件，语法格式如下：

```
<input type="file" name="文件域的名称"/>
```

使用文件域上传文件时，一定别忘记设置 form 表单信息提交的编码方式为 enctype="multipart/form-data"。如下面的代码：

```
<form action="" enctype="multipart/form-data">
    你的靓照：<input type="file" name="fileName"/>
</form>
```

上述代码呈现的结果如图 8.11 所示。

你的靓照：[　　　　　　] [浏览...]

图 8.11　文件域的呈现结果

8.5.2　能力目标

能使用 input 标记创建图像域、隐藏域以及文件域，并且掌握图像域、隐藏域以及文件域的使用方法。

8.5.3　任务驱动

1. 任务的主要内容

编写网页 example8_5.html，具体要求如下：

- 网页中有个表单，表单的处理程序是页面本身，表单中有 1 个文件域、1 个图像域与 1 个隐藏域；
- 文件域的名称为 myFile；
- 图像域的名称为 pic，地址设置为 logon.PNG；
- 隐藏域的名称为 flag，值为 0；
- 页面运行效果如图 8.12 所示。

2. 任务的代码模板

将下列 example8_5.html 中的【代码】替换为 HTML 代码。

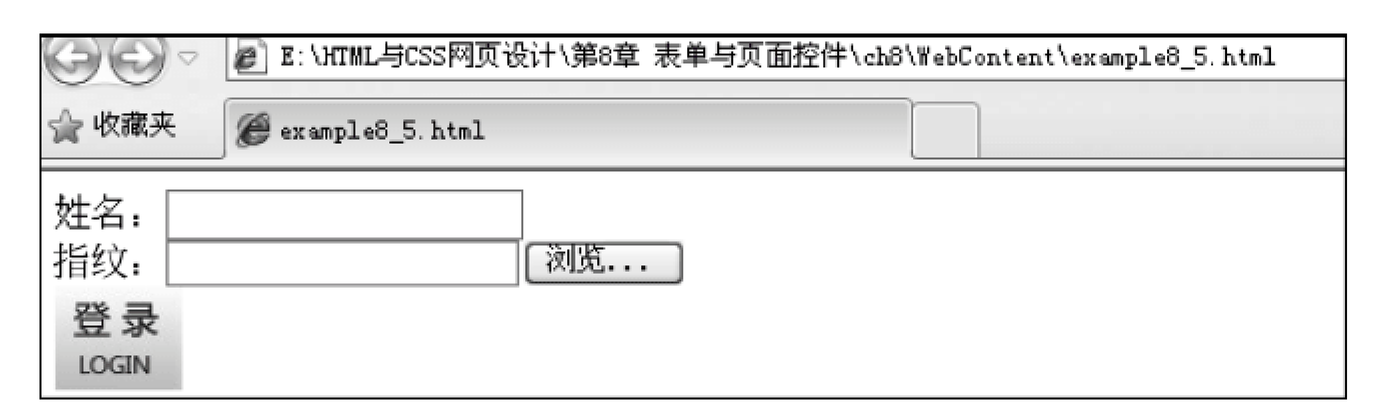


图 8.12　example8_5.html 页面运行效果

example8_5.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_5.html</title>
</head>
<body>
    <!--代码 1设置 form表单信息提交的编码方式为 multipart/form-data-->
    <form action=""【代码 1】>
        姓名：<input type="text" name="userName"/><br>
        <!--代码 2创建文件域 -->
        指纹：<input【代码 2】name="myFile"><br>
        <!--代码 3创建图片域 -->
        <input【代码 3】name="pic" src="logon.PNG"/>
        <!--代码 4创建隐藏域 -->
        <input【代码 4】name="flag" value="0"/>
    </form>
</body>
</html>
```

3. 任务小结或知识扩展

一个程序需要处理多个表单提交的请求时，一般情况下，需要在每个表单中添加一个名称相同值不同的隐藏域，处理程序根据隐藏域的值就可以判断是哪个表单提交的请求。

4. 代码模板的参考答案

【代码 1】：enctype="multipart/form-data"
【代码 2】：type="file"
【代码 3】：type="image"
【代码 4】：type="hidden"

8.5.4　实践环节

编写一个网页 practice8_5.html，具体要求如下：

- 网页中表单的处理程序是页面本身；
- 表单中有 1 个文件域、1 个图像域以及 3 个按钮(类型分别是“提交”按钮、“重置”按钮、普通按钮)；

- 文件域的名称为 yourFile；
- 图像域的名称为 yourPic，地址设置为 logon. PNG；
- 普通按钮具有关闭窗口的功能；
- 页面运行效果如图 8. 13 所示。

图 8. 13　practice8_5. html 页面运行效果

8. 6　列表与文本区

8.6.1　核心知识

1. 下拉列表

下拉列表在正常状态下只显示一个选项，单击下拉按钮后才会看到全部选项，语法格式如下：

```
<select name="下拉列表的名称">
    <option value="选项值 1" selected>选项 1 显示内容
    <option value="选项值 2" >选项 2 显示内容
    …
</select>
```

其中，选项值是提交给服务器的值，而选项显示内容才是真正在页面中要显示的。selected 表示此选项在默认状态下是选中的，selected 在下拉列表中只能有一个，因为一个下拉列表中只能有一个选项被选中。如下面的代码：

```
<select name="cities">
    <option value="beijing" selected>北京
    <option value="shanghai" >上海
    <option value="guangzhou" >广州
    <option value="shenzhen" >深圳
</select>
```

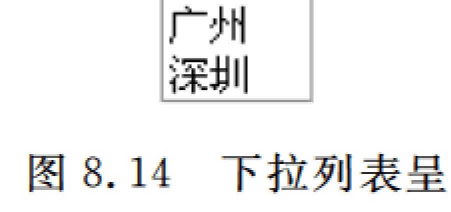


图 8. 14　下拉列表呈现结果

上述代码呈现的结果如图 8. 14 所示。

2. 滚动列表

滚动列表和下拉列表的实现形式是一样的，只是外观上不太一样。滚动列表在页面中可以显示多条信息，一旦超出这些信息，在列表右侧就会出现滚动条，拖动滚动条就可以看

到所有的选项内容，语法格式如下：

```
<select name="滚动列表的名称" size="显示的项数" multiple>
    <option value="选项值 1" selected>选项 1 显示内容
    <option value="选项值 2" >选项 2 显示内容
    …
</select>
```

其中，size 用来设定滚动列表在页面中最多显示的项数，当超出这个值时就会出现滚动条。multiple 表示滚动列表可以进行多项选择。选项值是提交给服务器的值，而选项显示内容才是真正在页面中要显示的。selected 表示此选项在默认状态下是选中的。如下面的代码：

```
<select name="cities" size="2" multiple >
    <option value="beijing" selected>北京
    <option value="shanghai" selected>上海
    <option value="guangzhou" >广州
    <option value="shenzhen" >深圳
</select>
```

上述代码呈现的结果如图 8.15 所示。

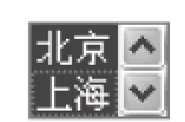


图 8.15　滚动列表呈现结果

3. 文本区

当用户需要输入多行文本时，就应该使用文本区而不是文本框了。文本区和其他表单控件不一样，它使用的是 textarea 标记而不是 input 标记，语法格式如下：

```
<textarea name="文本区的名称" cols="列数" rows="行数"></textarea>
```

其中，cols 用于设定文本区的列数，也就是其宽度；rows 用于设定文本区的行数，也就是高度值，当文本区的内容超出这一范围时就会出现滚动条。

8.6.2　能力目标

理解并掌握下拉列表、滚动列表以及文本区的使用方法。

8.6.3　任务驱动

1. 任务的主要内容

编写网页 example8_6.html，具体要求如下：

- 网页中有 1 个下拉列表、1 个滚动列表以及 1 个文本区；
- 下拉列表的名称为 college，选项显示内容有“利比亚国家大学、伊朗伊斯兰大学、叙利亚国际大学、美国与叙利亚联合大学”，选项“美国与叙利亚联合大学”默认被选中；
- 滚动列表的名称为 country，显示项数为 2，选项显示内容有“利比亚、伊朗、叙利亚、美国”，选项“美国”与“叙利亚”默认被选中；
- 页面运行效果如图 8.16 所示。



图 8.16　example8_6.html 页面运行效果

2. 任务的代码模板

将下列 example8_6.html 中的【代码】替换为 HTML 代码。

example8_6.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example8_6.html</title>
</head>
<body>
选择你的大学：
    <!--代码 1 定义下拉列表 college -->
    【代码 1】
        <option value="0">利比亚国家大学
        <option value="1">伊朗伊斯兰大学
        <option value="2">叙利亚国际大学
        <option value="3" selected>美国与叙利亚联合大学
    </select>
    <br>
选择你喜欢的国家：
    <!--代码 2 定义滚动列表 country,滚动列表允许多选,显示项数为 2 -->
    【代码 2】
        <option value="a" selected>叙利亚
        <option value="b" selected>美国
        <option value="c">伊朗
        <option value="d">利比亚
    </select>
    <br>
    备注：
    <!--代码 3 定义文本区,文本区的行数为 10,列数为 10 -->
    【代码 3】请慎重选择!</textarea>
</body>
</html>
```

3. 任务小结或知识扩展

从 example8_6. html 中可以看出文本区中的默认内容是写在<textarea>与</textarea>标记之间的。当文本区中的内容超出了文本区行数的大小时，在文本区右侧就会出现滚动条。

在下拉列表中使用 selected 把多个选项选中，仅有最后一个使用 selected 的选项被选中。因为下拉列表只能选中一个选项。

4. 代码模板的参考答案

【代码 1】：<select name="college">
【代码 2】：<select name="country" size="2" multiple>
【代码 3】：<textarea rows="10" cols="10">

8.6.4 实践环节

把 example8_6. html 中的代码：

```
<select name="college">
    <option value="0" >利比亚国家大学
    <option value="1" >伊朗伊斯兰大学
    <option value="2" >叙利亚国际大学
    <option value="3" selected>美国与叙利亚联合大学
</select>
```

改成如下代码：

```
<select name="college">
    <option value="0" selected>利比亚国家大学
    <option value="1" selected>伊朗伊斯兰大学
    <option value="2" selected>叙利亚国际大学
    <option value="3" selected>美国与叙利亚联合大学
</select>
```

保存代码后，再运行 example8_6. html 页面，查看下拉列表中哪个选项被选中。

8.7 小　　结

网页中的表单是由 form 标记定义的，<form>标记里的 action 属性是必不可少的，它指定了表单的处理程序。表单提交方法有 get 和 post。get 方法将表单内容以附件形式放在 URL 地址后面，不安全；post 方法将用户在表单中输入的数据包含在表单主体中，一起提交给服务器，该方法没有信息长度的限制，也比较安全。

表单对象 input 通过 type 属性指定表单对象的类型，如 type 的值为 text 时，input 表单对象代表的是文本框类型。

下拉列表、滚动列表以及文本区等控件不是通过 input 表单对象定义的。select 标记定义下拉列表与滚动列表控件，textarea 标记定义文本区控件。

习　题　8

1. 下列哪一项表示的不是按钮(　　)。

   A. type="submit"　　B. type="reset"

   C. type="image"　　D. type="button"

2. 若要产生一个 4 行 30 列的多行文本域,以下方法中,正确的是(　　)。

   A. <input type="text" rows="4" cols="30" name="txtintrol">

   B. <textarea rows="4" cols="30" name="txtintro">

   C. <textarea rows="4" cols="30" name="txtintro"></textarea>

   D. <textarea rows="30" cols="4" name="txtintro"></textarea>

3. 用于设置文本框显示宽度的属性是(　　)。

   A. size　　B. maxlength　　C. value　　D. length

4. 下面对表单的说法错误的是(　　)。

   A. 表单在 Web 页面中用来给访问者填写信息,从而能采集客户端信息,使页面具有交互信息的功能

   B. 当用户填写完信息后单击普通按钮做提交(submit)操作

   C. 一个表单用<form></form>标志来创建

   D. action 属性的值是指处理程序的程序名(包括网络路径、网址或相对路径)

5. 下面说法错误的是(　　)。

   A. 在 html 语言中,标记<input>具有重要的地位,它能够将浏览器中的控件加载到 html 文档中,该标记既有开始标记,又有结束标记

   B. <input type="text">是设定一个单行的文本输入区域

   C. size 属性指定控件宽度,表示该文本输入框所能显示的最大字符数

   D. maxlength 属性表示该文本输入框允许用户输入的最大字符数

6. 下面对于按钮的说法中正确的是(　　)。

   A. 按钮可分为普通按钮、"提交"按钮和"重置"按钮

   B. <input type="button">表示这是个提交到服务器的按钮

   C. <input type="RESET">表示这是个普通按钮

   D. name 属性用来指定按钮页面显示的名称

7. 下面说法中错误的是(　　)。

   A. <input type="checkbox " checked>,其中 checked 属性用来设置该复选框缺省时是否被选中

   B. <input type="hidden">表示一个隐藏区域。用户可以在其中输入某些要传送的信息

   C. <input type="password">表示这是个密码区域。当用户输入密码时,区域内将会显示"*"号

   D. <input type="radio">表示这是一个单选按钮

8. 下列说法中错误的是(　　)。
   A. <select></select>标志对用来创建一个菜单下拉列表框
   B. 下拉列表框中 multiple 属性不用赋值,直接加入标志中即可使用,加入了此属性后列表框就成了可多选的了
   C. <option>标志用来指定列表框中的一个选项
   D. 不可以设定输入多行的文本区的大小
9. 编写 1 个网页习题 8-9.html,具体要求如下:
   - 在习题 8-9.html 网页中有个 form 表单,表单处理程序为本页面程序,表单提交方式为"post",表单提交编码方式为"multipart/form-data";
   - 页面运行效果如图 8.17 所示。

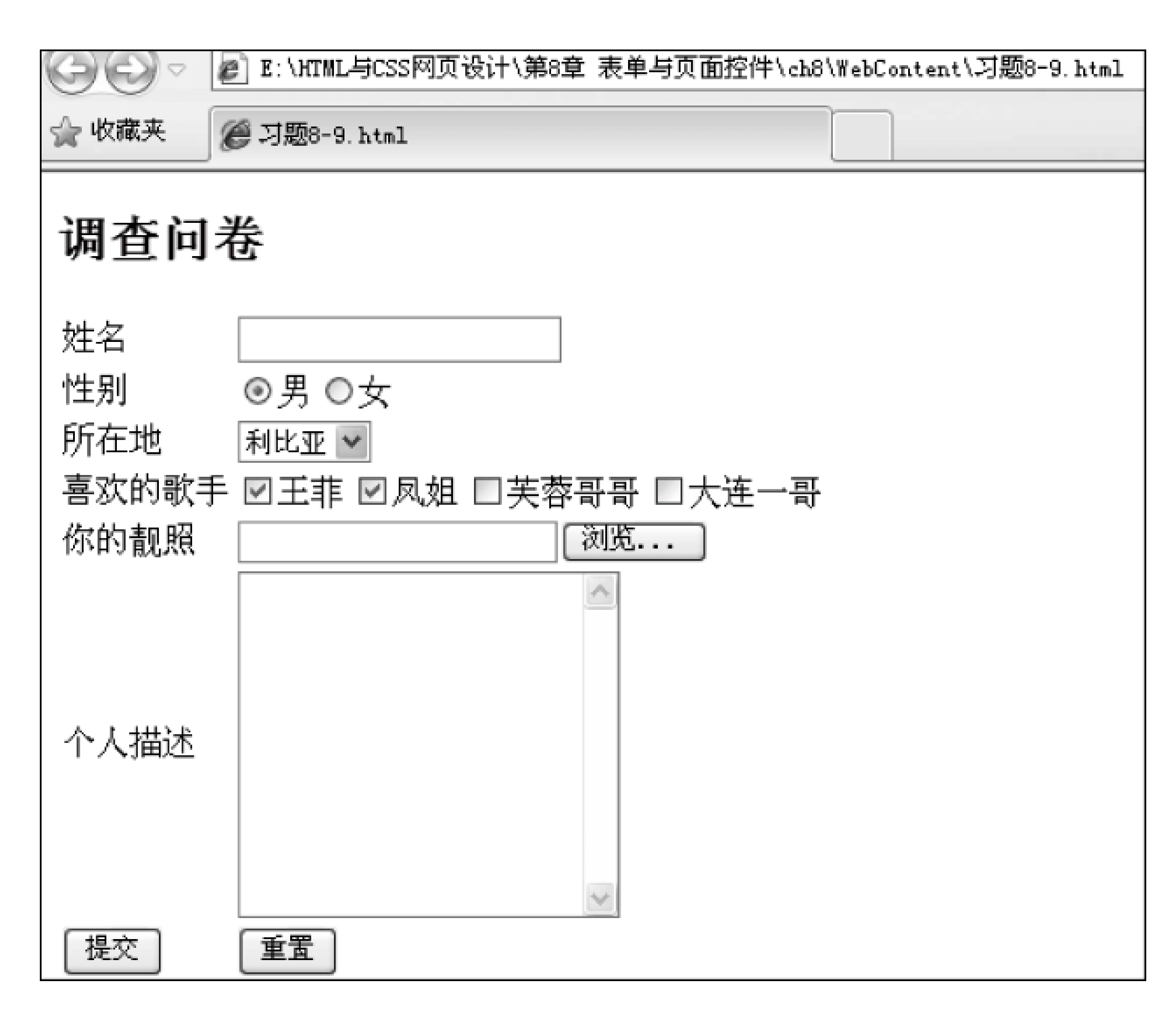


图 8.17　习题 8-9.html 页面运行效果

第 9 章

框架与浮动窗口

主要内容

- 框架集与框架
- 浮动框架
- 框架链接

我们在浏览网站时，经常通过网站的导航栏浏览网站中具体的网页内容，这些具体的网页内容通常在一个浏览器窗口中显示。这实际上是利用了框架技术制作了网站的导航栏。框架通常是由框架集和框架组成的。框架集就像一个容器，包括了框架，框架在框架集中是一个独立的区域块，用于独立显示一个网页内容。

9.1 框架集与框架

框架最大的用途就是导航，一组框架通常包括一个含有导航栏的框架和另一个显示网页内容的框架。

9.1.1 核心知识

框架页面的结构在框架集中设置，包括水平分割窗口、垂直分割窗口和嵌套分割窗口。

1. 水平分割窗口

水平分割窗口就是将页面按照水平方向切割成上下排列的多个窗口。与将一个表格分成几行类似，语法格式如下：

```
<frameset rows="frame 窗口 1 高度, frame 窗口 2 高度,…">
    <frame src="页面文件 1 地址"/>
    <frame src="页面文件 2 地址"/>
    …
</frameset>
```

其中，rows 可以取多个值，多个值之间用逗号分开，这些值可以是像素，也可以是百分比。例如，下面的整个窗口被水平分割成 3 个框架窗口。

frame 窗口 1
frame 窗口 2
frame 窗口 3

2. 垂直分割窗口

垂直分割窗口就是将页面按照垂直方向切割成左右排列的多个窗口。与将一个表格分成几列相似,语法格式如下:

```
<frameset cols="frame窗口1宽度, frame窗口2宽度, ...">
    <frame src="页面文件1地址"/>
    <frame src="页面文件2地址"/>
    ...
</frameset>
```

其中,cols可以取多个值,多个值之间用逗号分开,这些值可以是像素,也可以是百分比。例如,下面的整个窗口被垂直分割成4个框架窗口。

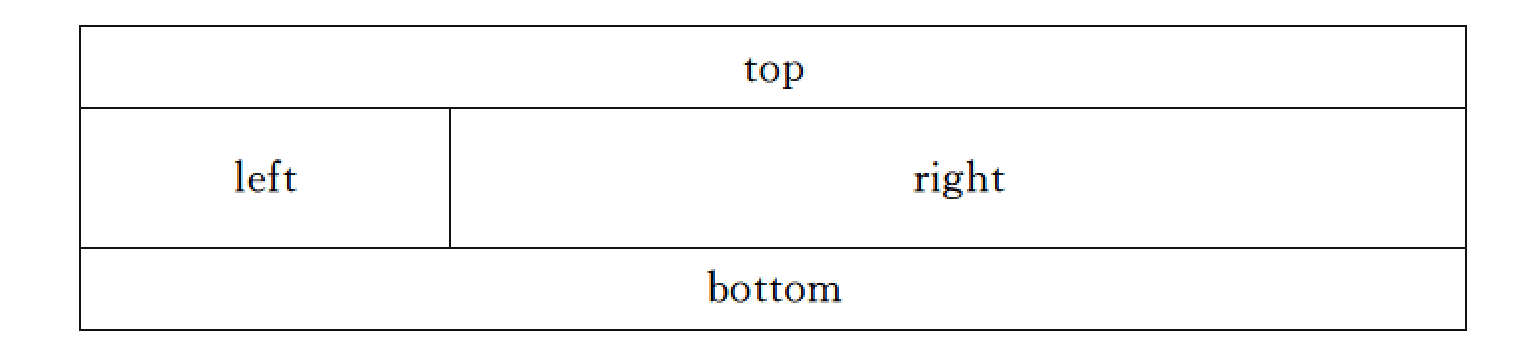
frame 窗口 1	frame 窗口 2	frame 窗口 3	frame 窗口 4

3. 嵌套分割窗口

嵌套分割窗口就是页面中既有水平分割的窗口,又有垂直分割的窗口。类似于将一个表格分成几行几列,语法格式如下:

```
<frameset rows="frame窗口1高度, frame窗口2高度, ...">
    <frame src="页面文件1地址"/>
    <frameset cols="frame窗口1宽度, frame窗口2宽度, ...">
        <frame src="页面文件2地址"/>
        <frame src="页面文件3地址"/>
        ...
    </frameset>
    ...
</frameset>
```

嵌套分割窗口是框架中最常见的一种方式,因为它可以很灵活地将窗口页面任意分割。例如,下面的整个窗口首先被水平分割成3个框架窗口,然后中间的窗口又被垂直分割成2个框架窗口。



4. 框架集的属性

我们可以通过frameborder让框架结构的边框显示或隐藏,也可以通过framespacing

属性调整框架结构边框的线条粗细，还可以通过 bordercolor 设置框架结构边框的颜色，语法格式如下：

```
<frameset frameborder="是否显示" framespacing="边框宽度" bordercolor="边框颜色">
   …
</frameset>
```

其中，frameborder 的值可以为 0、1 或 yes、no。如果为 0 或 no，那么边框隐藏；如果为 1 或 yes，那么边框显示。在 frameset 中设置的 frameborder 将会对整个框架集有效，而 frame 中设置的 frameborder 只对该框架有效。framespacing 的值默认情况下为 1 像素，该属性只能对框架集设置，对单个框架无效。bordercolor 属性也只能对框架集设置，对单个框架无效。

5. 框架的属性

每个框架窗口都可以通过一些属性设置自己的样式，语法格式如下：

```
<frame src="框架页面地址" name="框架名称" marginwidth="水平边距"
scrolling="是否显示滚动条" marginheight="垂直边距" noresize/>
```

每个框架窗口都有一个显示页面，该页面称为框架页面。框架页面的地址可以是绝对地址，也可以是相对地址，还可以是锚点链接的地址。为了便于页面的查找和链接，利用 name 属性为每个框架命名。框架窗口在默认情况下，会在右侧或下方显示滚动条，可以通过 scrolling 属性改变。scrolling 取值包括 yes、no 或 auto，其中，yes 表示滚动条一直显示，no 表示滚动条一直不显示，auto 是系统的默认值，根据内容自动调整(内容超出时就显示滚动条)。noresize 属性是让框架窗口的大小保持不变。

9.1.2　能力目标

理解并掌握框架集与框架的创建方法以及如何使用它们的属性设置不同的样式。

9.1.3　任务驱动

1. 任务的主要内容

编写网页 example9_1.html，具体要求如下：

- 页面中有个框架集，框架集把整个页面分割成上、下两部分，上部分高度为 200 像素，整个框架集不带滚动条；
- 上面框架的地址是 top.html；
- 下面框架里面又嵌套一个框架集，框架集把下面框架按垂直分割成左、右两部分，左部分宽度为 200 像素，整个框架集不带滚动条；
- 左侧框架的地址是 left.html，右侧框架的地址是 right.html；
- 所有框架不允许调整其大小，网页运行效果如图 9.1 所示。

2. 任务的代码模板

将下列 example9_1.html 中的【代码】替换为 HTML 代码。



图 9.1 example9_1.html 运行效果

example9_1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example9_1.html</title>
</head>
<!--代码 1 定义框架集,把页面按水平分割成上、下两部分,上部分高度为 200,框架集没有边框-->
    【代码 1】
<!--代码 2 定义框架,框架地址是 top.html,框架没有滚动条,不允许调整大小-->
        【代码 2】
<!--代码 3 定义框架集,把页面按垂直分割成左、右两部分,左部分宽度为 200,框架集没有边框-->
        【代码 3】
            <frame src="left.html" scrolling="no" noresize>
            <frame src="right.html" scrolling="no" noresize>
        </frameset>
    </frameset>
</html>
```

top.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>top.html</title>
</head>
<body>
```

```
    <embed src="top.swf" width="800" type="application/x-shockwave-flash" >
    </embed>
</body>
</html>
```

left.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>left.html</title>
</head>
<body bgcolor="PaleTurquoise">
    <table>
        <tr>
            <td>新 闻</td>
        </tr>
        <tr>
            <td>军 事</td>
        </tr>
        <tr>
            <td>视 频</td>
        </tr>
        <tr>
            <td>银 行</td>
        </tr>
    </table>
</body>
</html>
```

right.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>right.html</title>
</head>
<body bgcolor="Pink">
    <h2>购票网站瘫痪 垄断才是“真凶”</h2>
    <p>12306 火车订票网站在 1 月 9 日的点击量超过了 14 亿。据悉，目前主营铁路网上售票业务
       的是铁道部下属企业，属于垄断行业国企，有人称“肥水不流外人田”(1 月 10 日《第一财经日
       报》)。另外，据报该网站已烧钱数千万，却频现瘫痪、崩溃等现象 (1 月 10 日《21 世纪经济报
       道》)。
    <p>点击次数从日均 10 亿迅速攀升到 14 亿，似乎都是“人多惹的祸”。不过，当我们了解到车票
       这一目前最稀缺的公共资源被独家垄断经营时，就会认识到不仅是“人多惹的祸”，还有“垄
       断惹的祸”。
</body>
</html>
```

3. 任务小结或知识扩展

从 example9_1. html 的代码中,我们怎么没有看到 body 标记呢? 框架在框架集中是一个独立的区域块,用于独立显示一个网页内容。因此,框架不需要放在 body 标记里面,也就是说框架页面里面是不含有 body 标记的。

4. 代码模板的参考答案

```
【代码 1】: <frameset rows="200, * " frameborder="0">
【代码 2】: <frame src="top.html" scrolling="no" noresize>
【代码 3】: <frameset cols="200, * " frameborder="0" >
```

9.1.4 实践环节

在 example9_1. html 中的</head>后面加上<body>,在 example9_1. html 中的</html>前面加上</body>,保存代码后运行 example9_1. html 页面,查看运行结果。

9.2 浮动框架

浮动框架是一类很特别的框架,它是在浏览器窗口中嵌套的一个子窗口,整个页面不一定是框架页面,但包含一个框架窗口。另外,浮动框架不需要在框架集中。

9.2.1 核心知识

浮动框架<iframe>完全由程序设计者定义宽度和高度,并且可以放置在网页的任何位置,这样极大地拓展了框架页面的应用范围,语法格式如下:

```
<iframe src="浮动页面地址" width="浮动框架的宽" height="浮动框架的高"
    align="对齐方式" name="浮动框架的名称" scrolling="是否显示滚动条"
    frameborder="是否显示边框">
</iframe>
```

其中,一些属性和前面学习过的框架集与框架的属性类似,这里不再解释。

9.2.2 能力目标

理解并掌握浮动框架的创建方法以及如何使用它的属性设置不同的样式。

9.2.3 任务驱动

1. 任务的主要内容

编写网页 example9_2. html,具体要求如下:

- 在 example9_2. html 页面中有个<h2>级别的标题"浮动框架的学习",在标题的下方有个表格,该表格分 3 行,第 1 行第 1 列里面放置的是一个浮动框架;
- 浮动页面的地址是 leftiframe. html;
- 网页运行效果如图 9.2 所示。

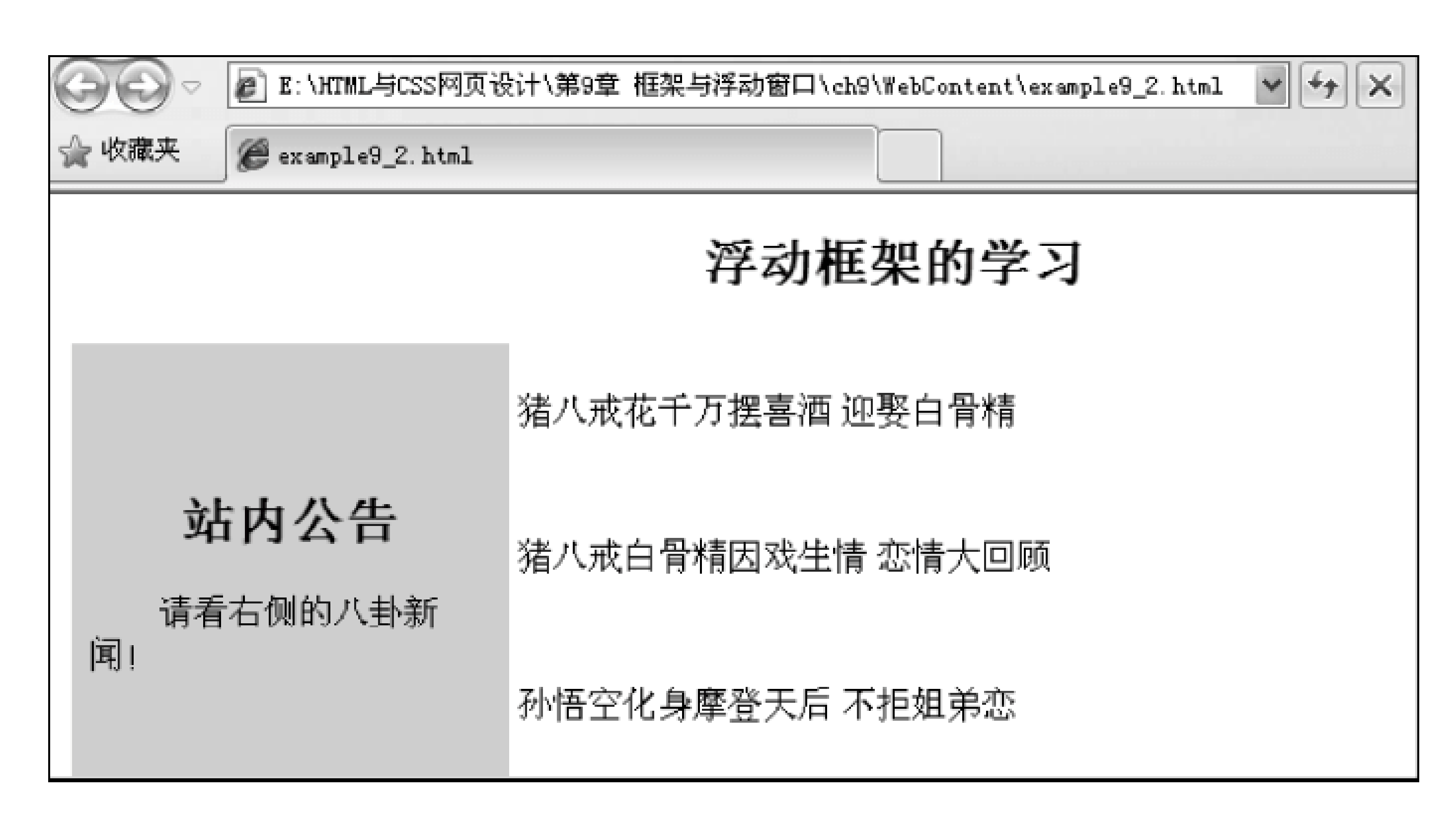


图 9.2　example9_2.html 运行效果

2. 任务的代码模板

将下列 example9_2.html 中的【代码】替换为 HTML 代码。

example9_2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example9_2.html</title>
</head>
    <body>
        <h2 align="center">浮动框架的学习</h2>
        <table>
            <tr>
                <td rowspan="3">
                <!--代码 1 定义浮动页面的地址为 leftiframe.html-->
                <!--代码 2 设置浮动框架的宽度和高度都为 200 -->
                    <iframe【代码 1】【代码 2】
                    align="left" name="left" scrolling="no" frameborder="no">
                    </iframe>
                </td>
                <td>
                    猪八戒花千万摆喜酒 迎娶白骨精
                </td>
            </tr>
            <tr>
                <td>
                    猪八戒白骨精因戏生情 恋情大回顾
                </td>
            </tr>
            <tr>
                <td>
                    孙悟空化身摩登天后 不拒姐弟恋
                </td>
```

```
            </tr>
        </table>
    </body>
</html>
```

leftiframe.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>leftiframe.html</title>
</head>
    <body bgcolor="PaleTurquoise">
        <marquee direction="up" behavior="alternate" scrolldelay="300">
            <h2 align="center">站内公告</h2>
            <p>     请看右侧的八卦新闻!
        </marquee>
    </body>
</html>
```

3. 任务小结或知识扩展

iframe 也是框架的一种形式，相当于在主浏览器窗口内嵌一个子窗口，内容自动打开。iframe 可以嵌在网页中的任意部分，也可以随意定义其大小。iframe 可以解决加载缓慢的第三方内容(如图标和广告等)的加载问题。

4. 代码模板的参考答案

【代码 1】: src="leftiframe.html"
【代码 2】: width="200" height="200"

9.2.4 实践环节

在网页 example9_2.html 中的表格的最后一个单元格里面添加一个浮动框架，浮动页面由读者自己定义。

9.3 框 架 链 接

框架链接与普通的链接有所不同。在框架链接中可以指定打开窗口的位置，从而实现在一个框架区域中，通过链接控制另一个框架区域的显示内容。

9.3.1 核心知识

1. 普通框架(frame)结构的链接

一般情况下，一个页面中会有一个框架窗口作为导航页面，在导航页面里面添加了对另一个框架内容的链接。它们通过 target 互相链接。下面通过一个实例来说明。

下面是框架集中的代码段：

```
…
<frameset rows="200,* ">
    <frame src="top.html" name="top"/>
    <frameset cols="300,* ">
        <frame src="left.html" name="left"/>
        <frame src="right.html " name="right"/>
    </frameset>
</frameset>
…
```

下面是 top 框架中的内容作为页面的导航页，代码段如下：

```
…
<a href="sportnews.html " target="right">体育新闻</a>
…
```

在该实例中，我们可以看到每个框架 frame 都有一个属性 name，这个 name 要和导航页面中超链接的 target 一致，如该实例中的 right。当我们单击“体育新闻”这个超链接时，就会把 sportnews. html 页面的内容显示在 right 框架中，也就是说在 right. html 页面显示的位置处显示 sportnews. html 页面的内容。name 的值是一个标识符，可以根据需要随便定义，但要和链接中的 target 值相同，才能实现框架链接。

2. 浮动框架的链接

浮动框架的链接和普通框架的链接类似。下面通过一个实例来说明。

实例代码段如下：

```
…
<iframe src="myiframe.html " width="500" height="200" name="Iframe" ></iframe>
…
<a href="youriframe.html " target="Iframe">企业简介</a>
…
```

在该实例中，我们可以看到浮动框架 iframe 有一个属性 name，这个 name 要和超链接的 target 一致，如该实例中的 Iframe。当我们单击“企业简介”这个超链接时，就会把 youriframe. html 页面的内容显示在浮动框架中，也就是说在 myiframe. html 页面显示的位置处显示 youriframe. html 页面的内容。name 的值是一个标识符，可以根据需要随便定义，但要和链接中的 target 值相同，才能实现浮动框架链接。

9.3.2　能力目标

理解并掌握框架链接的使用方法。

9.3.3　任务驱动

1. 任务的主要内容

编写网页 example9_3. html，具体要求如下。

- 将 example9_3. html 页面从上至下分为 top、page、bottom 3 个区域。page 是页面的主要内容，在 page 中将页面分为左右两部分，分别为 left、right。各部分的布局规

划如图 9.3 所示。

- 当单击 left 部分的超链接时,会把响应的网页内容显示在 example9_3.html 页面的 right 部分中。
- top 框架的页面地址是 topFrame.html,left 框架的页面地址是 leftFrame.html,right 框架的页面地址是 rightFrame.html,bottom 框架的页面地址是 bottomFrame.html。



图 9.3　example9_3.html 运行效果

2. 任务的代码模板

将下列 example9_3.html 与 leftFrame.html 中的【代码】替换为 HTML 代码。

example9_3.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example9_3.html</title>
</head>
    <frameset rows="100,400, * " frameborder="0">
        <frame src="topFrame.html" scrolling="no" name="top" noresize>
        <frameset cols="100, * " frameborder="0">
            <frame src="leftFrame.html" scrolling="no" name="left" noresize>
            <!--代码 1 定义一个框架,框架页面地址是 rightFrame.html,框架名称为 right,框架
              不允许滚动与调整大小 -->
            【代码 1】
```

```
        </frameset>
        <frame src="bottomFrame.html" scrolling="no" name="bottom" noresize>
    </frameset>
</html>
```

leftFrame. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>leftFrame.html</title>
</head>
<body bgcolor="PaleTurquoise">
    <table>
        <tr>
        <!--代码 2 定义超链接,链接地址是 rightFrame.html,链接页面显示在 right 框架中-->
        <td>
            【代码 2】style="text-decoration:none">个人简介</a>
        </td>
        </tr>
        <tr>
        <!--代码 3 定义超链接,链接地址是 gerentedian.html,链接页面显示在 right 框架中-->
        <td>
            【代码 3】style="text-decoration:none">个人特点</a>
        </td>
        </tr>
    </table>
</body>
</html>
```

topFrame. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>topFrame.html</title>
</head>
<body bgcolor="Wheat">
    <br><br>
    <center><font size=6 face=华文新魏>猪八戒的个人网站</font></center>
</body>
</html>
```

rightFrame. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>rightFrame.html</title>
```

```
</head>
<body bgcolor="Pink">
    <h2 align="center">个人简介</h2>
    <br>
    <table>
        <tr>
            <td><p>猪八戒,西游记中人物。法号悟能,是唐僧的二徒弟,原来是玉皇大帝的天蓬
                元帅,因调戏嫦娥被逐出天界,到人间投胎,却又错投猪胎,嘴脸与猪相似。他
                会变身术,能腾云驾雾,使用的兵器是九齿钉耙。唐僧西去取经路过云栈洞,
                猪八戒被孙悟空收服,八戒从此成为孙悟空的好帮手,一同保护唐僧去西天
                取经。八戒性格温和,憨厚单纯,力气大,但又好吃懒做,爱占小便宜,贪图女
                色,经常被妖怪的美色所迷,难分敌我。他对师兄的话言听计从,对师父忠心
                耿耿,为唐僧西天取经立下汗马功劳,是个被人们喜爱同情的喜剧人物。
            </td>
        </tr>
        <tr>
            <td>八戒性格温和,憨厚单纯,力气大,嘴巴甜。但好吃懒做,爱占小便宜,贪图女色,
            经常被妖怪的美色所迷惑,难分敌我。猪八戒贪吃贪睡,自私自利,好进谗言。他
            常常想捉弄人,但不是搬起石头来砸自己的脚就是作茧自缚。为什么连这样一个
            人也最终修成正果,被如来佛封为"净坛使者"。这与猪八戒的知错就改,能听取
            领导和同事的意见有重要联系(其实这与悟空的'督促'有很大关系),做到了悬崖
            勒马,甚至是浪子回头。在取经事业中的他对师兄的话还算得上言听计从,对师
            父忠心耿耿,总算是为西天取经立下汗马功劳,才有净坛使者的美好结果吧。
            </td>
        </tr>
    </table>
</body>
</html>
```

gerentedian.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>gerentedian.html</title>
</head>
<body>
<br>贪吃
<br>懒做
<br>贪睡
<br>心宽
<br>体胖
<br>好色
<br>心善
<br>忠诚
<br>老实
<br>胆小
</body>
</html>
```

bottomFrame.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>bottomFrame.html</title>
</head>
<body bgcolor="Wheat">
<br>
<center>版权归清华大学出版社所有</center>
</body>
</html>
```

3. 任务小结或知识扩展

不管是 frame 框架链接还是 iframe 浮动框架链接，都要注意一点：链接的 target 属性值和框架的 name 属性值必须一致。

4. 代码模板的参考答案

```
【代码 1】：<frame src="rightFrame.html" scrolling="no" name="right" noresize>
【代码 2】：<a href="rightFrame.html" target="right"
【代码 3】：<a href="gerentedian.html" target="right"
```

9.3.4 实践环节

编写一个网页 practice9_3.html，具体要求如下：

- 在 practice9_3.html 页面中首先有两个超链接，然后有个浮动框架；
- 当单击链接时，会把响应的网页内容显示在 practice9_3.html 页面的浮动框架中；
- 页面运行效果如图 9.4 所示。



图 9.4　practice9_3.html 运行效果

9.4 小　　结

frameset 创框架集，在框架集中使用 frame 创建框架窗口，框架可以嵌套另一个框架集。框架在框架集中是一个独立的区域块，用于独立显示一个网页内容。框架不需要放在 body 标记里面，也就是说框架页面里面是不含有 body 标记的。

浮动框架 iframe 是在浏览器窗口中嵌套的一个子窗口。浮动框架可以放在任意位置，也可以任意设置窗口大小。浮动框架主要解决加载缓慢的第三方内容（如图标和广告等）的加载问题。

习　题　9

1. 下面对框架的说法错误的是（　　）。
   A. 框架页面把浏览器窗口切割成几个独立的部分
   B. 设计框架页面时，<frame>标记和<frameset>标记用于定义框架网页的结构
   C. 由于框架网页的出现，从根本上改变了 HTML 文档的传统结构，因此在出现<frameset>标记的文档中，将不再使用<body>标记
   D. <frame>是用来划分窗框的，每一窗框由一个<frameset>标记所标示，<frameset>必须在<frame>范围中使用
2. 下面说法错误的是（　　）。
   A. 我们可以将窗口分割为几块，横向分用 rows 属性，纵向分用 cols 属性
   B. 框架可以嵌套
   C. <frameset cols="*,*,*">是将窗口横向分为三等分
   D. <frameset rows="50%,50%"> 是将窗口横向分为 2 个，各占 50%的显示区域
3. 下面说法中正确的是（　　）。
   A. 框架的各个窗口间不可以相互操作
   B. src 属性应该出现在<frameset>标记中
   C. <a href=url target=_blank>的作用是说明超链接的内容显示一个新窗口
   D. <a href=url target=right >的作用是说明超链接的内容显示在浏览器窗口的右侧
4. 框架与浮动框架有什么区别？

第 10 章

JavaScript

主要内容

- 在网页中添加 JavaScript 的方法
- JavaScript 基本语法
- 流程控制和函数
- JavaScript 对象
- JavaScript 对象模型
- 操作 HTML
- 表单验证

JavaScript 是一种描述性的脚本语言(Script Language),它由客户端浏览器解释执行,执行期间无须 Web 服务器,减轻了 Web 服务器的负担。

10.1 在网页中添加 JavaScript 的方法

10.1.1 核心知识

1. 嵌入使用

在网页代码中任何位置都可嵌入 JavaScript 代码,但建议嵌入到 head 标记中。示例代码如下:

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>JavaScript 嵌入学习</title>
    <script type="text/javascript">
        alert("第一次看到警告框很兴奋!");
    </script>
    </head>
    <body>
        好好学习 JavaScript 知识。
    </body>
</html>
```

2. 引入使用

当多个页面使用相同的JavaScript代码时，可以将共用的代码保存在以.js为扩展名的文件中，然后在页面中使用src属性引入外部js文件。示例代码如下：

myFirst.js

```
alert("被引入到页面中!");
```

JavaScript引入学习.html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>引入外部js文件</title>
    <script type="text/javascript" src="myFirst.js" charset="GBK">
    </script>
    </head>
    <body>
        好好学习JavaScript知识。
    </body>
</html>
```

10.1.2 能力目标

理解并掌握在网页中使用JavaScript的方法。

10.1.3 任务驱动

1. 任务的主要内容

编写网页example10_1.html，在网页中嵌入使用JavaScript代码连续两次弹出“How do you do?”。

2. 任务的代码模板

将下列example10_1.html中的【代码】替换为程序代码。

example10_1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<title>example10_1.html</title>
<!--嵌入JavaScript,使用alert两次弹出How do you do? -->
【代码1】
</head>
<body>
    先执行head里面的JavaScript,然后执行body部分。如果JavaScript程序是函数的话,只有函
    数被调用时,才执行JavaScript。
</body>
</html>
```

3. 任务小结或知识扩展

当浏览器载入网页body部分时，执行写在<body>...</body>里的JavaScript语句，

执行之后输出的内容就显示在网页中。有时候并不需要一载入 HTML 就运行 JavaScript，而是用户单击了 HTML 中的某个控件对象，触发了一个事件后，才需要调用 JavaScript。这时候，通常将这样的 JavaScript 放在 HTML 的<head>...</head>里。

将 JavaScript 代码写在外部文件中，可以提高 JavaScript 的复用性，减少代码维护的负担，不必将相同的 JavaScript 代码拷贝到多个 HTML 网页里，将来一旦程序有所修改，也只要修改.js 文件就可以，不用再修改每个用到这个 JavaScript 程序的 HTML 文件。

4. 代码模板的参考答案

【代码 1】:
```
<script type="text/javascript">
        alert("How do you do?");
        alert("How do you do?");
</script>
```

10.1.4 实践环节

把 example10_1.html 中的 JavaScript 程序写在外部的 JS 文件里，然后在 example10_1.html 页面中引用该 JS 文件。

10.2 JavaScript 基本语法

JavaScript 区分大小写，变量是弱类型(任意变量类型都是 var 声明)。在浏览器中逐条解释执行，如果某条语句出现错误，后面的语句将不再执行。

10.2.1 核心知识

1. 变量

使用 var 可以声明任意类型的变量，如：

```
var firstNumber=10;
```

2. 类型转换

JavaScript 是弱类型语言，变量的类型对应于其值的类型。可以对不同类型的变量执行运算，解释器强制转换数据类型，然后进行处理。如：数值与字符串相加，将数值强制转换为字符串；布尔值与字符串相加，将布尔值强制转换为字符串；数值与布尔值相加，将布尔值强制转换为数值。

字符串到数值的转换。如：parseInt(s)将字符串转换为整数，parseFloat(s)将字符串转换为浮点数，Number(s)将字符串转换为数字。parseInt 方法和 parseFloat 方法只对 string 类型有效，且需要是数字开头的字符串。

3. 运算符

(1) 赋值运算符

赋值运算符的运算规则及说明如表 10.1 所示。

表 10.1　赋值运算符

运算符	示例	说　　明	运算符	示例	说　　明
=	x=y;	把变量 y 的值赋给 x	*=	x*=y;	x=x*y;
+=	x+=y;	x=x+y;	/=	x/=y;	x=x/y;
-=	x-=y;	x=x-y;			

(2) 数学运算符

数学运算符的运算规则及说明如表 10.2 所示。

表 10.2　数学运算符

运算符	示　　例	说　　明
+	A=5+8　　//结果是 13 A="5"+8　//结果是"58"	如果操作数都是数字时执行加法运算,如果其中的操作数有字符串时,会执行连接字符串的运算
-	A=8-5	减法
*	A=8*5	乘法
/	A=8/5//结果是 1.6	除法
%	10%3=1	取余
++	++x 返回递增后的 x 值 x++返回递增前的 x 值	递增
--	--x 返回递减后的 x 值 x--返回递减前的 x 值	递减
-	如果 a 等于 5,则-a=-5	此运算符返回操作数的相反数

(3) 逻辑运算符

逻辑运算符的运算规则及说明如表 10.3 所示。

表 10.3　逻辑运算符

运算符	示　　例	说　　明
&&	expr1 && expr2	逻辑与(表达式 1 错误,表达式 2 不再运算)
||	expr1 || expr2	逻辑或(表达式 1 正确,表达式 2 不再运算)
!	!expr	逻辑非

(4) typeof 运算符

对变量或值调用 typeof 运算符将返回对应的值,typeof 运算符的运算规则及说明如表 10.4 所示。

表 10.4　typeof 运算符

示　　例	返回结果	说　　明
typeof(true)	boolean	变量或值是 boolean 类型
typeof(300)	number	变量或值是 number 类型
typeof('abc')	string	变量或值是 string 类型
typeof(null)	object	变量或值是一种引用类型或 null 类型
typeof(f)	function	变量是一个函数

10.2.2　能力目标

理解并掌握 JavaScript 变量的定义与使用，掌握运算符的使用方法，能灵活使用类型转换。

10.2.3　任务驱动

1. 任务的主要内容

编写网页 example10_2.html，在网页中嵌入使用 JavaScript，JavaScript 程序的具体要求如下：

- 声明变量 firstNumber，并把"I am a String"赋值给该变量；
- 声明变量 secondNumber，并把 300 赋值给该变量；
- 声明变量 thirdNumber，并把 firstNumber＋secondNumber 赋值给该变量；
- 使用 alert 弹出变量 thirdNumber 的值；
- 使用 typeof 运算符判断变量 firstNumber、secondNumber 以及 thirdNumber 的值类型，并使用 alert 分别弹出 typeof 的返回结果。

2. 任务的代码模板

将下列 example10_2.html 中的【代码】替换为程序代码。

example10_2.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example10_2.html</title>
<script type="text/javascript">
    var firstNumber="I am a String";
    var secondNumber=300;
    var thirdNumber=firstNumber+secondNumber;
    【代码 1】//弹出变量 thirdNumber 的值
    alert("firstNumber 的值类型："+【代码 2】);     //弹出变量 firstNumber 的值类型
    alert("secondNumber 的值类型："+【代码 3】);    //弹出变量 secondNumber 的值类型
    alert("thirdNumber 的值类型："+【代码 4】);     //弹出变量 thirdNumber 的值类型
</script>
</head>
<body>
</body>
</html>
```

3. 任务小结或知识扩展

（1）注释

为了程序的可读性，以及便于日后代码修改和维护，可以在 JavaScript 程序里为代码写注释。在 JavaScript 程序里，用"//"来表示单行注释。多行注释用"/*"表示开始，"*/"表示结束。注释示例代码如下：

```
aGoodIdea="Comment your code thoroughly.";        //这是单行注释
/*
这是多行注释 行一。
这是多行注释 行二。
*/
```

(2) 变量命名规则

以字母、下画线(_)或美元符号($)开头;余下的字符可以是下画线、美元符号或任何的字母、数字;不能有空格,大小写敏感;不能使用 JavaScript 中的关键字或保留字命名。

(3) 部分保留字

break、delete、function、return、typeof、case、do、if、switch、var、catch、else、in、this、void、continue、false、instanceof、throw、while、finally、new、true、with、default、for、null、try

4. 代码模板的参考答案

```
【代码 1】: alert(thirdNumber);
【代码 2】: typeof(firstNumber)
【代码 3】: typeof(secondNumber)
【代码 4】: typeof(thirdNumber)
```

10.2.4 实践环节

编写网页 practice10_2.html,在网页中嵌入使用 JavaScript,JavaScript 程序的具体要求如下:

- 声明变量 firstNumber,并把"300number"赋值给该变量;
- 声明变量 secondNumber,并把"number300"赋值给该变量;
- 使用 parseInt 将 firstNumber 转换为整数,并弹出转换后的值;
- 使用 parseInt 将 secondNumber 转换为整数,并弹出转换后的值。

10.3 流程控制与函数

10.3.1 核心知识

1. 流程控制语句

(1) if 条件语句

if 条件语句格式如下:

```
if(表达式){
    语句
}
```

或

```
if(表达式)){
    语句
}else{
    语句
}
```

（2）switch 条件语句

switch 条件语句格式如下：

```
switch(表达式){
    case case1:
        语句
        break;
    case case2:
        语句
        break;
    …
    default:
        default 语句
}
```

（3）for 循环语句

for 循环语句格式如下：

```
for(表达式 1;表达式 2;表达式 3){
    语句
}
```

（4）while 循环语句

while 循环语句格式如下：

```
while(表达式){
    语句
}
```

（5）do-while 循环语句

```
do{
    语句
}while(表达式)
```

（6）break/continue 语句

break 语句让执行语句从循环语句或其他程序块中跳出。continue 语句让执行语句跳过本次循环的剩余语句进入下一次循环。

2. 函数

把完成某个功能的一组语句常写成一个函数，函数的定义格式如下：

```
function 函数名([参数,参数]){
    函数体
}
```

function 是关键字，函数没有类型，参数也没有类型。如：

```
function gogo(obj){
    document.write("函数没有类型,参数也没有类型");
}
```

3. arguments 对象

函数可以接受任意个数的参数，通过 arguments 对象来访问。示例代码如下：

```
function say(){
    if(arguments[1] !="你好"){
        alert(arguments[0]);
    }else{
        alert(arguments[1]);
    }
    alert(arguments.length);                  //返回参数的个数
}
```

调用函数如下：

```
say("How are you?","你好");
```

4. 系统函数

JavaScript 提供了与任何对象无关的系统函数，这些函数不需要创建任何对象就可以直接使用。

（1）eval(字符串表达式)

该函数的功能是返回字符串表达式的值，如：

```
var test=eval("2+3");                         //test 的值为 5
```

（2）parseInt(字符串)

该函数的功能是把以数字开头的字符串转换为整数，如：

```
var test=parseInt("200.5abc");                //test 的值为 200
```

（3）parseFloat(字符串)

该函数的功能是把以数字开头的字符串转换为实数，如：

```
var test= parseFloat("200.5abc");             //test 的值为 200.5
```

（4）Number(字符串)

该函数的功能是把数字字符串转换为数字，字符串中有非数字字符则返回 NaN，如：

```
var test=Number("200.5abc");                  //test 的值为 NaN
```

10.3.2 能力目标

理解并掌握流程控制语句与函数的用法。

10.3.3 任务驱动

1. 任务的主要内容

编写网页 example10_3.html，在网页中嵌入使用 JavaScript 程序打印出九九乘法表，网页运行效果如图 10.1 所示。

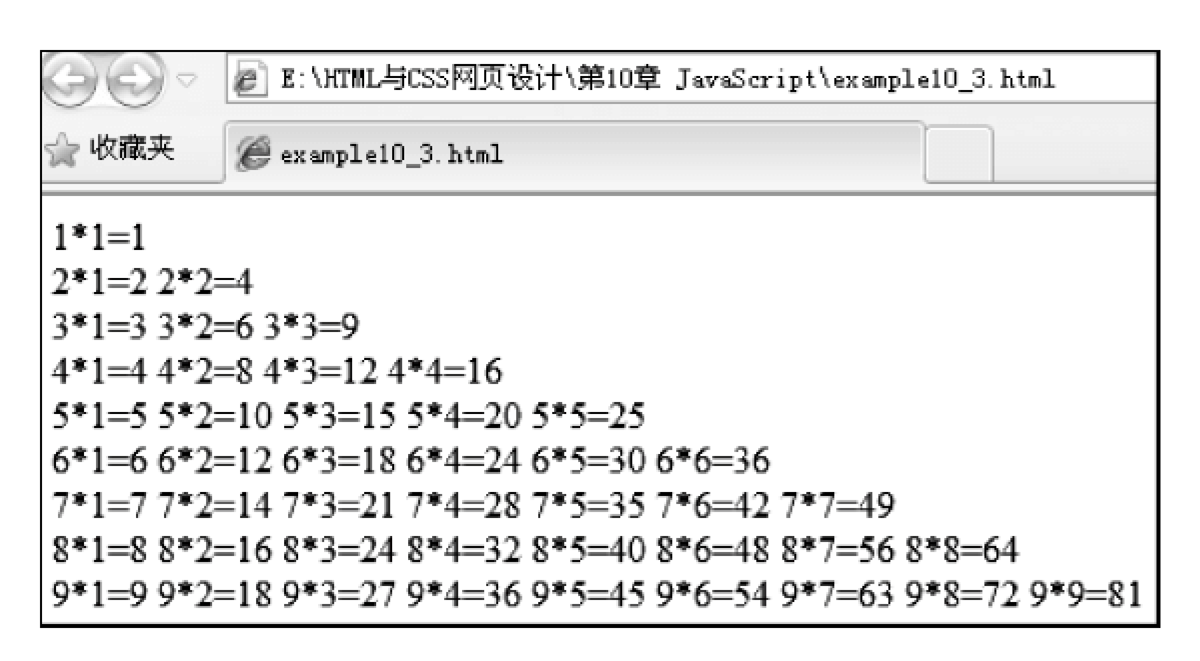


图 10.1　九九乘法表

2. 任务的代码模板

将下列 example10_3. html 中的【代码】替换为程序代码。

example10_3. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example10_3.html</title>
<script type="text/javascript">
【代码 1】{//第一层 for 循环
    【代码 2】{//第二层 for 循环
        document.write(i+"*"+j+"="+i*j+" ");    //在页面输出
    }
    document.write("<br>");
}
</script>
</head>
<body>
</body>
</html>
```

3. 任务小结或知识扩展

（1）在链接中调用函数

用户单击链接时，即调用函数，格式如下：

```
<a href="javascript:函数">…</a>
```

在链接中调用函数的示例代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>链接调用函数</title>
<script type="text/javascript">
function gogo(){
```

```
        alert("被链接调用的函数");
    }
    </script>
    </head>
    <body>
        <a href="javascript:gogo()">链接调用函数</a>
    </body>
    </html>
```

(2) 由事件触发调用函数

触发事件调用函数,格式如下:

```
事件="函数"
```

触发事件调用函数的示例代码如下:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>事件触发调用函数</title>
<script type="text/javascript">
function gogo(param){
    alert(param);
}
</script>
</head>
<body>
    <form action="">
        <input type="button" value="单击我" onclick="gogo('o(∩_∩)o哈哈~ ')"/>
    </form>
</body>
</html>
```

4. 代码模板的参考答案

【代码 1】: for(var i=1;i<=9;i++)
【代码 2】: for(var j=1;j<=i;j++)

10.3.4 实践环节

把 example10_3.html 中的打印九九乘法表程序写在 JavaScript 函数中,并在页面的 body 中使用超链接调用该函数。

10.4 JavaScript 对象

一个 JavaScript 对象中可包含若干属性和方法。属性是描述对象的状态,对象用“.”运算符访问属性。方法是描述对象的行为动作,对象用“.”运算符调用方法。

10.4.1　核心知识

1. 对象创建

使用 new 关键字来创建对象，如：

```
var oStringObject=new String();
```

如果构造函数无参数，则不必加括号。

2. JavaScript 内部对象

1）数组（Array）对象

（1）创建数组

数组的创建，示例代码如下：

```
var myArray=new Array();              //新建一个长度为 0 的数组
var myArray=new Array(100);           //新建一个长度为 100 的数组
var myArray=new Array(1,2,3);         //新建一个指定长度的数组，并赋初值
```

数组长度不固定，赋值即可改变长度。数组的主要属性 length，返回数组长度的整数值。

（2）数组的常用方法

reverse()方法：将 JavaScript 数组对象内容反转。

concat()方法：将两个或更多数组组合在一起，如：

```
var newArray=tmpArray.concat(tmpArray);
```

join 方法：返回一个将数组所有元素用指定符号连在一起的字符串，如：

```
var newString=tmpArray.join(".");
```

pop()方法：移除数组中的最后一个元素并返回该元素。

shift()方法：移除数组中的第一个元素并返回该元素。

slice()方法：返回数组的一部分，如：

```
var newArray=tmpArray.slice(1,3);
```

（3）数组的使用

数组的使用，示例代码如下：

```
<script type="text/javascript">
    var myArray=new Array();
    for(var i=0;i<5;i++){
        myArray[i]=i;
    }
    for(var j=0;j<myArray.length;j++){
        alert(myArray[j]);
    }
</script>
```

2）日期（Date）对象

Date 对象可以用来表示任意的日期和时间。

（1）创建 Date 对象

必须使用 new 运算符创建一个 Date 对象。示例代码如下：

```
var date=new Date("July 8,2012");   //2012 年 7月 8 日
var date=new Date(2012,7,8);        //2012 年 8月 8 日
var date=new Date("2012/7/8") ;     //2012 年 7月 8 日
var date=new Date(Milliseconds);    //自 1970 年 1月 1 日以来的毫秒数创建的日期对象
var date=new Date();                //当前系统的时间对象
```

（2）获取日期和时间的方法

getYear()：返回年数。

getMonth()：返回当月号数(比实际值小 1)。

getDate()：返回当日号数。

getDay()：返回星期几(0 表示周日)。

getHours()：返回小时数。

getMinutes()：返回分钟数。

getSeconds()：返回秒数。

getTime()：返回毫秒数。

（3）设置日期和时间的方法

setYear()：设置年数。

setMonth()：设置当月号数(set6 表示 7 月)。

setDate()：设置当日号数。

setHours()：设置小时数。

setMinutes()：设置分钟数。

setSeconds()：设置秒数。

setTime()：设置毫秒数。

（4）Date 对象的使用

Date 对象的使用，示例代码如下：

```
<script type="text/javascript">
    var date=new Date("2050/12/25");
    document.write("2050 年的圣诞节是星期"+date.getDay()+"<br>");
    var datenow=new Date();   //得到当前日期对象
    var mills=date-datenow;   //两个 Date 对象相减得到两个日期的时间间隔(单位是毫秒)
    document.write("2050 年的圣诞节距离现在还有"+mills+"毫秒<br>");
</script>
```

3）String 对象

（1）创建 String 对象

字符串对象的创建，示例代码如下：

```
var firstString="This is a string";
var secondString=new String("This is a string");
```

String 对象的主要属性 length，返回字符串长度的整数值。

(2) String 对象的常用方法

charAt(i)：返回指定索引位置处的字符，索引从 0 开始。

concat(str)：连接字符串。

indexOf(str)：返回 String 对象内第一次出现子字符串的字符位置(从左到右查找)。

lastIndexOf(str)：返回 String 对象中子字符串最后出现的位置。

replace(str1,str2)：返回将 str1 替换为 str2 后的字符串。

split(separator, limit)：将字符串以 separator 作为分割符切割成多个子字符串，并将它们作为一个数组返回；如果有 limit 参数则返回数组的 limit 个元素。

substring(start,end)：返回一个指定位置之间的子字符串，不包括 end。

toLowerCase()：返回一个字符串，字符串中的字母被转换为小写字母。

toUpperCase()：返回一个字符串，字符串中的字母被转换为大写字母。

(3) String 对象的使用

String 对象的使用，示例代码如下：

```
<script type="text/javascript">
    var firstString="This is a string";
    for(var i=0;i<firstString.length;i++){
        alert(firstString.charAt(i));
    }
</script>
```

4) Math 对象

Math 对象是个全局对象，使用时不需要创建。

(1) Math 对象的属性

LN10：10 的自然对数。

LN2：2 的对数。

PI：圆周率。

SQRT1_2：1/2 的平方根。

SQRT2：2 的平方根。

(2) Math 对象的常用方法

abs(x)：返回 x 的绝对值。

ceil(x)：返回大于等于 x 的最小整数。

floor(x)：返回小于等于 x 的最大整数。

round(x)：舍入到最近整数。

sqrt(x)：返回 x 的平方根。

random()：返回 0～1 的随机数。

(3) Math 对象的使用

Math 对象的使用，示例代码如下：

```
<script type="text/javascript">
    alert(Math.SQRT2);
    alert(Math.random());
</script>
```

10.4.2 能力目标

理解并掌握 JavaScript 对象的创建与使用。

10.4.3 任务驱动

1. 任务的主要内容

编写网页 example10_4. html，在网页中嵌入使用 JavaScript，JavaScript 程序的具体要求如下：

- 声明变量 dateString，并把日期字符串"2012-08-08"赋值给该变量；
- 使用 substring 方法获得"2012-08-08"日期的年份、月份与日，并打印在网页上。

2. 任务的代码模板

将下列 example10_4. html 中的【代码】替换为程序代码。

example10_4. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example10_4.html</title>
<script type="text/javascript">
    var dateString="2012-08-08";
    var year=【代码 1】          //获得年份
    var month=【代码 2】         //获得月份
    var day=【代码 3】           //获得日
    document.write("年份为: "+year+"<br>");
    document.write("月份为: "+month+"<br>");
    document.write("日为: "+day+"<br>");
</script>
</head>
<body>
</body>
</html>
```

3. 任务小结或知识扩展

在 JavaScript 中不必专门销毁对象来释放内存，当没有针对某对象的引用时，该对象被自动废除。也可以把对象的所有引用设置为 null 来废除对象。

4. 代码模板的参考答案

```
【代码 1】: dateString.substring(0,4);
【代码 2】: dateString.substring(5,7);
【代码 3】: dateString.substring(8,10);
```

10.4.4 实践环节

编写网页 practice_4. html，在网页中定义一个 JavaScript 函数，功能是去除字符串开头

及末尾的空格，并使用超链接调用该函数（思路：使用 charAt 和 substring 去除头尾的空格）。

10.5 JavaScript 对象模型

10.5.1 核心知识

1. 浏览器对象模型

浏览器对象是提供独立于内容而与浏览器窗口进行交互的对象。浏览器对象模型如图 10.2 所示。



图 10.2 浏览器对象模型

2. 窗口（window）对象

1）打开新窗口

使用 open()方法可打开一个新窗口，示例代码如下：

```
var winObj=open("target.html", "target_1","width=500,height=300,scrollbars=no");
```

open 方法有 3 个参数：第一个代表要载入新窗口页面的 URL；第二个代表新窗口的名称；第三个代表新窗口的属性，多个属性间用逗号隔开。

2）对话框（与用户交互）方法

（1）alert()

该方法的功能是弹出一个提示框。示例代码如下：

```
<script type="text/javascript">
   alert("请点击确定按钮!");
</script>
```

上述代码运行效果如图 10.3 所示。

(2) prompt(message,defaultText)

该方法的功能是弹出可以输入信息的文本框,第一个参数代表用户输入信息的提示,第二个参数代表文本框的默认值。示例代码如下:



图 10.3　具有"确定"按钮的对话框

```
<script type="text/javascript">
    prompt("What's your name?","chenheng");
</script>
```

上述代码运行效果如图 10.4 所示。

(3) confirm(message)

该方法的功能是弹出对话框,提示确认信息。示例代码如下:

```
<script type="text/javascript">
    if(confirm("真的删除吗?")){
        //单击确定后的操作
    }else{
        //单击取消后的操作
    }
</script>
```

上述代码运行效果如图 10.5 所示。



图 10.4　用户输入对话框

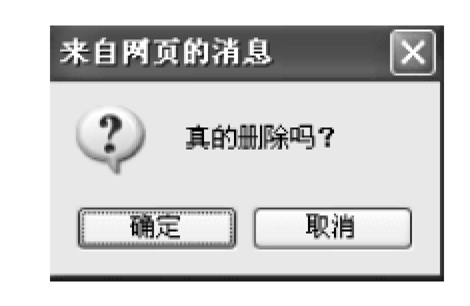


图 10.5　确认对话框

3. history 对象

history 对象记录着浏览器所浏览过的每一个页面,这些页面组成了一个历史记录列表。它有如下 3 个主要方法。

- forward():将历史记录向前移动一个页面。
- back():将历史记录向后移动一个页面,在网页中经常使用该方法提供一个"返回"的功能。
- go():转向历史记录中指定地址,使用此方法需要一个参数,参数可以是正负整数或字符串。如果参数是字符串,那么浏览器就会搜索列表,找到最接近当前页面位置且 URL 地址中含有此字符串的页面,然后转到该页面。

history 对象的使用,示例代码如下:

```
history.go(-3);         //向后返回三个访问过的页面
histroy.go(3);          //向前返回三个访问过的页面
history.back();         //与 history.go(-1);功能相同
history.forward();      //与 history.go(1);功能相同
```

4. location 对象

window 对象的 location 属性可以直接改变 URL 地址：

```
window.location="http://www.baidu.com";
```

或

```
location="http://www.baidu.com";
```

还可以使用 location 对象的 href 属性或 replace(URL)方法改变 URL 地址：

```
location.href="http://www.baidu.com";
```

或

```
location.replace("http://www.baidu.com");
```

可以使用 location 对象的 href 属性清空页面：

```
location.href="about:blank";          //清空页面
```

5. document 对象

1）forms 集合

在同一个页面上有多个表单，通过 document.forms[]数组获得这些表单对象要比使用表单名称方便得多。

2）getElementById(id)方法

该方法的功能是获得指定 id 值的表单控件对象。

3）getElementsByName(name)方法

该方法的功能是获得指定 name 值的表单控件对象，返回的是对象数组。

4）获取表单对象的方法

获取表单对象的方法如下：

```
document.forms[0];                  //通过 forms 对象的索引
document.forms["myForm"];           //通过 forms 对象和表单名称
document.myForm;                    //通过表单名称
```

5）document 对象的使用

document 对象的使用，示例代码如下：

```
<html>
    <head>
    <title>The Document Object</title>
    <script type="text/javascript">
        function testMethod() {
            //1、通过表单获取控件
            var name_1=document.forms[0].nameTest;
            alert("name_1="+name_1.value);
            var name_2=document.forms["myForm"].nameTest;
            alert("name_2="+name_2.value);
            var name_3=document.myForm.nameTest;
```

```
            alert("name_3=" +name_3.value);
            var name_4=document.myForm.elements["nameTest"];
            alert("name_4=" +name_4.value);
            //2、通过 name 获取控件
            var name_5=document.getElementsByName("nameTest");
            alert("name_5[0]=" +name_5[0].value);
            //3、通过 id 获取控件
            var id_1=document.getElementById("idTest");
            alert("id_1=" +id_1.value);
        }
    </script>
    </head>
    <body>
        <form name="myForm">
            通过 name 获取:<input type="text" name="nameTest" value="firstTextValue">
            <br>
            <br>通过 id 获取:<input type="text" id="idTest" value="secondTextValue">
            <br><input type="button" value="Click Me" onclick="testMethod()">
        </form>
    </body>
</html>
```

10.5.2 能力目标

理解并掌握 JavaScript 对象模型(浏览器对象、窗口对象、history 对象、location 对象以及 document 对象)的使用方法。

10.5.3 任务驱动

1. 任务的主要内容

编写网页 example10_5.html,在网页中嵌入使用 JavaScript,程序的具体要求如下:

- 在页面中有表单控件对象——文本框和"提交"按钮;
- 在 JavaScript 程序中编写一个函数 getTextValue(),函数的功能是分别使用 getElementsByName 和 getElementById 获得文本框的值,并使用 alert 弹出该值;
- 单击"提交"按钮,触发 onclick 事件调用函数 getTextValue();
- 网页运行效果如图 10.6 所示。



图 10.6　example10_5 的运行效果

2. 任务的代码模板

将下列 example10_5.html 中的【代码】替换为程序代码。

example10_5.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>Insert title here</title>
<script type="text/javascript">
    function getTextValue(){
        var name1=【代码 1】 //通过 getElementsByName 获得名为 userName 的控件对象
        var name2=【代码 2】 //通过 getElementById 获得 id 为 name 的控件对象
        var value1=【代码 3】 //通过 name1 获得文本框的值
        var value2=【代码 4】 //通过 name2 获得文本框的值
        alert("通过 getElementsByName 获得的值:"+value1);
        alert("通过 getElementById 获得的值:"+value2);
    }
</script>
</head>
<body>
    <form action="">
        输入姓名:<input type="text" name="userName" id="name"/><br>
        <input type="button" value="提交" onclick="getTextValue()"/><br>
    </form>
</body>
</html>
```

3. 任务小结或知识扩展

getElementsByName 和 getElementById 的区别：在一个 HTML 文档中，每个元素都可以设置 id 和 name 属性。其中 id 属性是唯一属性，不可以重复，一个 id 值只能对应一个元素；而 name 属性是可以重复的，一个 name 值可以对应一组元素。所以，使用 getElementById 时返回的只有一个被选中的 HTML 元素；而使用 getElementsByName 时返回的是一个由多个 HTML 元素组成的数组（哪怕 HTML 页面中只有一个符合要求的元素，返回的也是数组）。getElementsByName 多用于在 form 表单中选取一组 checkbox 或 radio。

4. 代码模板的参考答案

```
【代码 1】: document.getElementsByName("userName");
【代码 2】: document.getElementById("name");
【代码 3】: name1[0].value;
【代码 4】: name2.value;
```

10.5.4 实践环节

编写网页 practice10_5.html，在网页中嵌入使用 JavaScript，JavaScript 程序的具体要求如下。

- 使用 prompt("","")获取用户输入的字符串，直到实现输入"STOP"时停止。
- 打印所有的输入：其他字符使用绿色，"STOP"使用红色。

10.6 操作HTML

10.6.1 核心知识

1. 事件处理

通常把光标或热键的动作称为事件。由光标或热键引发的一连串程序的动作,称为事件驱动。对事件进行处理的程序或函数称为事件处理程序。

(1) 窗口或页面的事件处理

窗口或页面的事件处理如表10.5所示。

表10.5 窗口或页面的事件处理

事　件	说　　明	事　件	说　　明
onBlur	当前元素失去焦点时触发	onLoad	页面内容完成装载时触发
onFocus	当某个元素获得焦点时触发	onUnload	当前页面被退出或重置时触发

(2) 键盘或鼠标的事件处理

键盘或鼠标的事件处理如表10.6所示。

表10.6 键盘或鼠标的事件处理

事　　件	说　　明
onClick	当鼠标单击时触发
onDblClick	当鼠标双击时触发
onMouseDown	当按下鼠标时触发
onMouseMove	当鼠标移动时触发
onMouseOut	当鼠标离开某对象范围时触发
onMouseOver	当鼠标移动到某对象范围的上方时触发
onMouseUp	当鼠标按下后松开鼠标时触发
onKeyPress	当键盘上的某个键被按下并且释放时触发
onKeyDown	当键盘上某个键被按下时触发
onKeyUp	当键盘上某个键被按放开时触发

(3) 表单元素的事件处理

表单元素的事件处理如表10.7所示。

表10.7 表单元素的事件处理

表 单 元 素	主 要 事 件
button(按钮)	onClick onBlur onFocus
checkbox(复选框)	onClick onBlur onFocus
file(上传文件)	onClick onBlur onFocus
password(密码框)	onBlur onFocus onSelect
radio(单选按钮)	onClick onBlur onFocus
select(列表)	onFocus onBlur onChange
text(文本框)	onClick onBlur onFocus onChange
textarea(文本区)	onClick onBlur onFocus onChange

(4) 事件处理模型

在 JavaScript 中对事件的处理程序通常由函数完成,事件="函数名",如:

```
<html>
    <head>
    <title>The Document Object</title>
    <script type="text/javascript">
        function testMethod() {
            alert("事件处理模型");
        }
    </script>
    </head>
    <body>
        <form name="myForm">
            <input type="button" value="Click Me" onclick="testMethod()" >
        </form>
    </body>
</html>
```

2 表单元素

(1) 表单元素的通用属性与方法

form 属性:获取该表单控件所属的表单对象。

name 属性:获取或设置表单控件的名称。

type 属性:获取表单控件的类型。

value 属性:获取和设置表单控件的值。

focus 方法:让表单控件对象获得焦点。

blur 方法:让表单控件对象失去焦点。

表单元素的通用属性及方法的示例代码如下:

```
<html>
    <head>
    <title>form</title>
    <script type="text/javascript">
        //显示属性
        function test() {
            var text_name=document.myForm.text_name;
            alert("text_name.form=" +text_name.form.name
                +"\ntext_name.name=" +text_name.name
                +"\ntext_name.type=" +text_name.type
                +"\ntext_name.value=" +text_name.value
                +"\ntext_name.defaultValue=" +text_name.defaultValue);
        }
        //获得焦点
        function do_focus() {
            document.myForm.text_name.focus();
        }
        //失去焦点
        function do_blur() {
```

```
            document.myForm.text_name.blur();
        }
    </script>
    </head>
    <body>
        <form name="myForm">
            <input type="text"  name="text_name"  value="textValue"><br>
            <input type="button" name="button_1" value="显示属性" onclick="test()">
              <br>
            <input type="button" name="button_2" value="获得焦点" onclick="do_focus()">
              <br>
            <input type="button" name="button_3" value="失去焦点" onclick=
              "do_blur()">
        </form>
    </body>
</html>
```

(2) 文本框

value 属性：获得文本框的值，值是字符串类型。

defaultValue 属性：获得文本框的默认值，值是字符串类型。

readonly 属性：只读，文本框中的内容不能修改。

focus 方法：获得焦点，即获得鼠标光标。

blur 方法：失去焦点。

select 方法：选中文本框内容，突出显示输入区域。

文本框的示例代码如下：

```
<html>
<head>
<title>文本框求和</title>
<script type="text/javascript">
function add(){
    var sum=0;
    var text_1Value=document.forms[0].text_1.value;
    var text_2Value=document.forms[0].text_2.value;
    sum=Number(text_1Value)+Number(text_2Value);
    document.forms[0].text_3.value=sum;
}
</script>
</head>
<body>
    <form name="form1" method="post" action="">
        <input type="text" name="text_1" value=""><br>
        <input type="text" name="text_2" value=""><br>
        <input type="text" name="text_3" value=""><br>
        <input type="button" value="求和" onclick="add()">
    </form>
</body>
</html>
```

(3) 复选框

checked 属性：复选框是否被选中，选中为 true，未选中为 false。

value 属性：设置或获取复选框的值。

复选框的示例代码如下：

```
<html>
<head>
<title>复选框反选</title>
<script type="text/javascript">
    function unSelect() {
        var n=document.forms[0].check.length;//得到复选框的个数
        for ( var i=0;i<n;i++) {
            if (document.forms[0].check[i].checked) {
                document.forms[0].check[i].checked=false;
            } else {
                document.forms[0].check[i].checked=true;
            }
        }
    }
</script>
</head>
<body>
    <form action="">
        <input type="checkbox" name="check" value="0" />aaa<br><input
            type="checkbox" name="check" value="1" />bbb<br><input
            type="checkbox" name="check" value="2" />ccc<br><input
            type="checkbox" name="check" value="3" />ddd<br><input
            type="checkbox" name="check" value="4" />eee<br><input
            type="checkbox" name="check" value="5" />fff<br><input
            type="button" value="反选" onclick="unSelect()">
    </form>
</body>
</html>
```

(4) 单选按钮

checked 属性：单选按钮是否被选中，选中为 true，未选中为 false。

value 属性：设置或获取单选按钮的值。

单选按钮的示例代码如下：

```
<html>
<head>
<title>弹出单选按钮的值</title>
<script type="text/javascript">
    function gg() {
        var n=document.forms[0].sex.length;
        for ( var i=0;i<n;i++) {
            if (document.forms[0].sex[i].checked) {
                alert(document.forms[0].sex[i].value);
            }
```

```
            }
        }
    </script>
    </head>
    <body>
        <form action="">
            <input type="radio" name="sex" value="male" onclick="gg()" />男
            <input type="radio" name="sex" value="female" onclick="gg()" />女
        </form>
    </body>
    </html>
```

(5) 下拉列表

length 属性：选项个数。

selectedIndex 属性：当前被选中选项的索引。

options 属性：所有的选项组成一个数组，options 表示整个选项数组，第一个选项为 options[0]，第二个为 options[1]，其他以此类推。

option 的 value 属性：<option>标记中 value 所指定的值。

option 的 text 属性：显示于界面中的文本，即<option>...</option>之间的部分。

增加选项：每个选项都是一个 option 对象，可以创建 option 对象，然后添加到 select 的末尾。如：select.options[select.length]=new Option(text,value);。

下拉列表的示例代码如下：

```
<html>
    <head>
    <title>下拉列表的应用</title>
    <script type="text/javascript">
        function gg() {
            var opObject=document.forms[0].elements["cities"];  //获得列表对象
            for ( var i=0;i<opObject.options.length;i++) {  //使用 options 属性
                if (opObject.options[i].selected) {     //找到被选中的选项
                    alert(opObject.options[i].value);   //弹出选中选项的值
                    alert(opObject.options[i].text);    //弹出选中选项的文本
                }
            }
            //为列表新增选项
            opObject.options[opObject.length]=new Option("新增"+1, "new"+1);
        }
    </script>
    </head>
    <body>
        <form action="">
            <select name="cities" onchange="gg()">
                <option value="daLian">大连</option>
                <option value="beiJing">北京</option>
                <option value="shangHai">上海</option>
                <option value="guanZhou">广州</option>
            </select>
```

```
        </form>
    </body>
</html>
```

10.6.2　能力目标

理解并掌握事件处理的方法以及如何使用 JavaScript 程序操作 HTML 元素。

10.6.3　任务驱动

1. 任务的主要内容

编写网页 example10_6.html，在网页中嵌入使用 JavaScript，程序的具体要求如下：

- 在页面中有多个文本框，每个文本框后面都有一个对应的复选框；
- 在 JavaScript 程序中编写一个函数 getSum()，该函数的功能是把多个文本中输入的数字进行求和(只有复选框被选中的文本框才进行求和)；
- 页面运行效果如图 10.7 所示。



图 10.7　多个文本框求和

2. 任务的代码模板

将下列 example10_6.html 中的【代码】替换为程序代码。

example10_6.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example10_6.html</title>
<script type="text/javascript">
    function getSum(){
        var n=document.forms[0].check.length;//得到复选框的个数
        var sum=0;
        for(var i=0;i<n;i++){
            if(【代码 1】){      //判断复选框是否被选中
                var textData=document.forms[0].data[i].value;   //得到文本框的值
```

```
                    sum=sum+Number(textData);
                }
            }
            alert(sum);
        }
</script>
</head>
<body>
<form action="">
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="text" name="data" value="">
    <input type="checkbox" name="check" value=""><br>
    <input type="button" value="求和" onclick="getSum()">
</form>
</body>
</html>
```

3. 任务小结或知识扩展

使用 JavaScript 操作 HTML，主要操作的是表单控件对象（文本框、复选框、单选按钮、下拉列表等）。

4. 代码模板的参考答案

【代码 1】：`document.forms[0].check[i].checked`

10.6.4 实践环节

把 example10_6.html 中的函数功能修改为把多个文本中输入的数字进行求和（只有复选框不被选中的文本框才进行求和）。

10.7 表单验证

用户在表单中输入的内容提交到服务器之前，在客户端利用表单控件产生的事件，运用脚本(JavaScript)，验证用户输入数据的有效性。

10.7.1 核心知识

1. 正则表达式

表单验证常用的正则表达式如下：

验证邮政编码：/^\d{6}$/

验证身份证号码：/^\d{15}$|^\d{18}$|^\d{17}[xX]$/

验证电子邮箱地址：/^\w+((-\w+)|(\.\w+))*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)*\.[A-Za-z0-9]+$/

验证数字或英文字母：/^[a-z0-9]+$/

验证日期格式：/((^((1[8-9]\d{2})|([2-9]\d{3}))([-\/\._])(10|12|0?[13578])([-\/\._])(3[01]|[12][0-9]|0?[1-9])$)|(^((1[8-9]\d{2})|([2-9]\d{3}))([-\/\._])(11|0?[469])([-\/\._])(30|[12][0-9]|0?[1-9])$)|(^((1[8-9]\d{2})|([2-9]\d{3}))([-\/\._])(0?2)([-\/\._])(2[0-8]|1[0-9]|0?[1-9])$)|(^([2468][048]00)([-\/\._])(0?2)([-\/\._])(29)$)|(^([3579][26]00)([-\/\._])(0?2)([-\/\._])(29)$)|(^([1][89][0][48])([-\/\._])(0?2)([-\/\._])(29)$)|(^([2-9][0-9][0][48])([-\/\._])(0?2)([-\/\._])(29)$)|(^([1][89][2468][048])([-\/\._])(0?2)([-\/\._])(29)$)|(^([2-9][0-9][2468][048])([-\/\._])(0?2)([-\/\._])(29)$)|(^([1][89][13579][26])([-\/\._])(0?2)([-\/\._])(29)$)|(^([2-9][0-9][13579][26])([-\/\._])(0?2)([-\/\._])(29)$))/

正则表达式的应用示例代码如下：

```
<html>
<head>
<title>正则表达的应用</title>
<script type="text/javascript">
//验证 Email
function checkEmail(){
    var exp=/^\w+((-\w+)|(\.\w+))*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)*\.
            [A-Za-z0-9]+$ /;
    if(!exp.test(document.forms[0].email.value)){
        alert("Email 格式错误!");
        document.forms[0].email.focus();
        return false;
    }else{
        alert("Email 格式正确!");
        return true;
    }
}
</script>
</head>
<body>
    <form action="">
```

```
        Mail: <input type="text" name="email" /><br>
        <input type="button" value="提交" onclick="return checkEmail()" />
    </form>
</body>
</html>
```

2. 表单验证实例

验证密码域：不能为空，长度要大于等于6，只能是字母或数字。代码如下：

```
<html>
<head>
<title>表单验证实例</title>
<script type="text/javascript">
function valid(form){
    //验证非空
    if(form.pass.value.length==0){
        alert("Please enter a password");
        form.pass.focus();
        return false;
    }
    //验证长度
    if(form.pass.value.length <6){
        alert("Password must be at least 6 characters");
        form.pass.focus();
        return false;
    }
    var exp=/^[a-z0-9]+$/;
    //验证格式
    if(!exp.test(form.pass.value)){
        alert("Password contains illegal characters");
        form.pass.focus();
        return false;
    }
    alert("OK password");
    return true;
}
</script>
</head>
<body>
    <form action="">
        Enter your password:
        <input name="pass" type="password"/>
        <input type="button" value="submit password" onclick="return valid
                    (this.form)"/>
    </form>
</body>
</html>
```

10.7.2　能力目标

理解并掌握表单验证的常用方法。

10.7.3　任务驱动

1. 任务的主要内容

编写网页 example10_7. html，在网页中嵌入使用 JavaScript，程序的具体要求如下：

- 在页面中有表单控件对象——文本框和“验证 ID”按钮；
- 在 JavaScript 程序中编写一个函数 checkID()，函数的功能是验证文本框中输入的身份证号码的合法性(考虑 15 位和 18 位身份证)；
- 单击“验证 ID”按钮，触发 onclick 事件调用函数 checkID()；
- 网页运行效果如图 10. 8 所示。



图 10. 8　身份证号码的验证

2. 任务的代码模板

将下列 example10_7. html【代码】替换为程序代码。

example10_7. html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example10_7.html</title>
<script type="text/javascript">
    function checkID(){
        var idValue=document.forms[0].indentity.value;
        var exp=/^\d{15}$|^\d{18}$|^\d{17}[xX]$/;
        if(【代码 1】){//代码 1 为合法的判断条件
            alert("身份证号码合法有效");
            return true;
        }else{
            alert("身份证号码无效");
            document.forms[0].indentity.focus();
            return false;
```

```
            }
        }
</script>
</head>
<body>
    <form action="">
        输入您的身份证号码：<input type="text" name="indentity"/><br>
        <input type="button" value="验证 ID" onclick="checkID()"/>
    </form>
</body>
</html>
```

3. 任务小结或知识扩展

表单验证的优点：在客户端验证，速度较快；减轻了服务器端的压力。

4. 代码模板的参考答案

【代码 1】：exp.test(idValue)

10.7.4 实践环节

制作一个用户注册页面 practice10_7.html，具体要求如下：

- 要有常用的登录账号、密码、确认密码、姓名、身份证号码(只考虑 18 位的身份证)、出生年月日、住址、邮编、E-mail 等输入区域(可以自己设定)；
- 自己设定验证规则，在提交时检验是否符合要求，(alert)出非法的输入，并把焦点返回要输入的控件对象；
- 根据出生年月日判断身份证号码是否合法(只考虑 18 位身份证，验证日期就可以)；
- 其他验证：登录账号只能是字母或数字且以字母开头；密码要在 8 位以上且需要有字母和数字之外的字符；出生年月日的格式为 yyyy-mm-dd；邮编为 6 位数字；E-mail 的基本格式验证即可。

10.8 小　　结

在网页中添加 JavaScript 的方法有两种：在 head 标记中嵌入使用和引入外部 JS 文件。

JavaScript 程序区分大小写，变量是弱类型即任意变量类型都是使用 var 声明。在浏览器中逐条解释执行，如果某条语句出现错误，后面的语句将不再执行。

JavaScript 提供了与任何对象无关的系统函数，不需要创建任何对象就可以直接使用这些函数，如：eval()、parseInt()、parseFloat()、Number()等。

JavaScript 提供了 Array、Date、Math 和 String 等内部对象。JavaScript 通常使用函数对表单控件对象产生的事件进行处理。

使用 JavaScript 在客户端对表单数据进行有效性验证是一项极其重要又复杂的工作，因为它需要尽可能全面地考虑数据的多样性，才能正确验证。

习　题　10

1. (　　)对象表示浏览器的窗口,可用于检索关于该窗口状态的信息。
   A. document　B. window　C. location　D. history
2. (　　)对象表示给定浏览器窗口中的 HTML 文档,用于检索关于文档的信息。
   A. document　B. window　C. screen　D. history
3. (　　)方法要求窗口显示刚刚访问的前一个页面。
   A. back()　B. go()　C. display()　D. view()
4. 有关变量的命名规则,下列说法中错误的是(　　)。
   A. 以字母、下画线(_)或美元符号($)开头
   B. 首字符之外的字符可以是下画线、美元符号或任何的字母、数字
   C. 不能有空格,不区分大小写
   D. 不能使用 JavaScript 中的关键字或保留字命名
5. 单击按钮触发的事件是(　　)。
   A. onClick　B. onFocus　C. onChange　D. onLoad
6. 页面加载时产生的事件是(　　)。
   A. onClick　B. onFocus　C. onChange　D. onLoad

第 11 章

CSS



- CSS 的语法
- 在网页中添加 CSS 的方法
- 选择符分类
- 伪类及伪类对象
- 常见 CSS 属性的设置方法

CSS 是英文 Cascading Style Sheet 的缩写，又称为“层叠样式表”，简称为样式表。它是 W3C 定义的标准，一种用来为结构化文档（如 HTML 文档）添加样式（字体、间距和背景等）的计算机语言。

11.1 CSS 的语法

11.1.1 核心知识

CSS 的语法由三部分构成：选择符（selector）、属性（property）和属性值（value）。语法格式如下：

```
选择符 {
    属性:值
}
```

选择符用来指定针对哪个 HTML 标签应用样式表，任何一个 HTML 标签都可以是一个 CSS 的选择符。如：

```
body {
    color: blue
}
```

其中，body 是选择符，color 是属性，blue 是属性值。该规则表示在网页 body 标签里的内容为蓝色。如果你想为选择符指定多个样式，在属性之间要用分号加以分隔。下面的选择符 p 就包含两个样式，一个是对齐方式居中，另一个字体颜色为红，当中用分号分隔开。

```
p {
    text-align: center;
    color: red
}
```

11.1.2　能力目标

理解并掌握 CSS 的语法结构。

11.1.3　任务驱动

1. 任务的主要内容

编写网页 example11_1.html，具体要求如下：

- 使用 CSS 设置网页的背景色为 LightBLue(淡蓝色)，背景色属性名为 background-color；
- 网页运行效果如图 11.1 所示。



图 11.1　example11_1.html 运行效果

2. 任务的代码模板

将下列 example11_1.html 中的【代码】替换为程序代码。

example11_1.html

```
<html>
<style type="text/css">
    /* 代码 1 使用 CSS 设置 body 的背景色 (background-color)为 LightBLue */
    【代码 1】
</style>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_1.html</title>
</head>
<body>
    第一次使用 CSS 设置网页的样式,感觉真神奇啊!
</body>
</html>
```

3. 任务小结或知识扩展

可以将相同的属性和属性值赋予多个选择符(组合)。选择符之间用逗号分隔。如：

```
h1,h2,h3,h4,h5,h6
{
    /* 字体颜色为蓝色 */
    color: blue
}
```

上面的例子是将所有正文标题(<h1>到<h6>)的字体颜色都变成蓝色。"/ * "和" * /"之间的内容为 CSS 的注释,但要注意不要将注释嵌入到选择符语句里面。

4. 代码模板的参考答案

```
【代码 1】: body {
            background-color: LightBLue
        }
```

11.1.4 实践环节

使用 CSS 设置 example11_1.html 页面的背景图片,背景图片属性及属性值写法为:

```
background-image:url(图片路径)
```

图片由读者自行选择。

11.2 在网页中添加 CSS 的方法

11.2.1 核心知识

CSS 在网页中按其位置可以分为三种:内嵌样式、内部样式和外部样式。

1. 内嵌样式(Inline Style)

内嵌样式是把样式代码写在标记里面的,使用 style 作为属性,样式语句作为属性值。内嵌样式只对所在的标记有效。如:

```
<p style="font-size:20pt;color:red">
    这个 style 定义<p></p>里面的文字是 20pt 字号,字体颜色是红色。
</p>
```

2. 内部样式(Internal Style Sheet)

内部样式是使用<style>标记把样式代码写在 HTML 的<head></head>里面的。内部样式只对所在的网页有效。如:

```
<html>
<head>
<style type="text/css">
    h1 {
        border-width:1;
        text-align:center;
        color:red
    }
</style>
</head>
<body>
    <h1>这个标题使用了 style。</h1>
</body>
</html>
```

3. 外部样式(External Style Sheet)

将样式代码写在一个以.css 为后缀的 CSS 文件里,然后在每个需要用到这些样式的网页里引用这个 CSS 文件。通过 HTML 的 link 元素将外部的样式文件连接到网页里。如:

```
<link rel="stylesheet" href="mystyle.css" type="text/css" />
```

其中,rel 和 type 属性表明这是一个样式文件,href 属性指定了外部样式文件的相对地址。外部的样式文件不能含有任何像<head>或<style>这样的 HTML 标记,样式表文件仅仅由样式规则或声明组成。

11.2.2 能力目标

理解并掌握在网页中添加 CSS 的方法。

11.2.3 任务驱动

1. 任务的主要内容

编写网页 example11_2.html,具体要求如下:

- 在 example11_2.html 页面中有个<h2>级别的标题"正在学习在网页中如何添加 CSS";
- 使用内嵌样式方法将<h2>标题的文字颜色设置成红色。

2. 任务的代码模板

将下列 example11_2.html 中的【代码】替换为程序代码。

example11_2.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_2.html</title>
</head>
<body>
    <h2【代码 1】>正在学习在网页中如何添加 CSS</h2>
</body>
</html>
```

3. 任务小结或知识扩展

使用外部样式,相对于内嵌样式和内部样式,有以下优点。

(1) 样式代码可以复用。一个外部 CSS 文件,可以被很多网页共用。

(2) 便于修改。如果要修改样式,只需要修改 CSS 文件,而不需要修改每个网页。

(3) 提高网页显示的速度。如果样式写在网页里,会降低网页显示的速度;如果网页引用一个 CSS 文件,这个 CSS 文件已经在缓存区(其他网页早已经引用过它),网页显示的速度就比较快。

因此,在实际开发中一般使用外部样式,不推荐使用内嵌样式和内部样式。

4. 代码模板的参考答案

【代码 1】: style="color: red"

11.2.4 实践环节

分别使用内部样式和外部样式将 example11_2. html 页面中<h2>标题的文字颜色设置成红色。

11.3 选择符的分类

CSS 的选择符分为：普通选择符、类选择符和 id 选择符。

11.3.1 核心知识

1. 普通选择符

任意的 HTML 标记都可以作为选择符，这样的选择符称为普通选择符。其样式仅仅作用在选择符指定的 HTML 标记上。如：

```
p {
    color:red
}
```

2. 类选择符

HTML 标记名加上类名，中间用“.”分开，类名供该 HTML 标记的 class 属性使用。希望<p>有两种样式，一种是居中对齐，一种是居右对齐。那么可以写成如下的样式：

```
p.right {
    text-align:right
}
p.center {
    text-align:center
}
```

其中，right 和 center 就是两个 class。在网页中可以引用这两个 class 设置段落的对齐方式。示例代码如下：

```
<p class="center">这一段居中显示。</p>
<p class="right">这一段是居右显示。</p>
```

如果省略 HTML 标记名只写“.类名”表示这个类名适用于所有的 HTML 标记的 class 属性，这种选择符称为通用类选择符。

3. id 选择符

HTML 标记名加上 id 名，中间用“#”分开，id 名供该 HTML 标记的 id 属性使用。如：

```
p# svp {
    font-size:12pt
}
```

其中,svp 是个 id 选择符的名字,在网页中可以引用这个 id 选择符设置<p>的样式。示例代码如下:

```
<p id="svp">这一段话的字体大小为 12pt。</p>
```

如果省略 HTML 标记名只写"#id 名"表示这个 id 名适用于所有的 HTML 标记的 id 属性,这种选择符称为通用 id 选择符。

11.3.2 能力目标

理解并掌握多种选择符的定义与使用方法。

11.3.3 任务驱动

1. 任务的主要内容

编写网页 example11_3.html,具体要求如下:

- 页面中有 h1～h4 标记的文字;
- 对 h1 与 h2 标记使用通用类选择符设置样式;
- 对 h3 标记使用通用 id 选择符设置样式;
- 对 h4 标记使用普通选择符设置样式;
- 网页运行效果如图 11.2 所示。

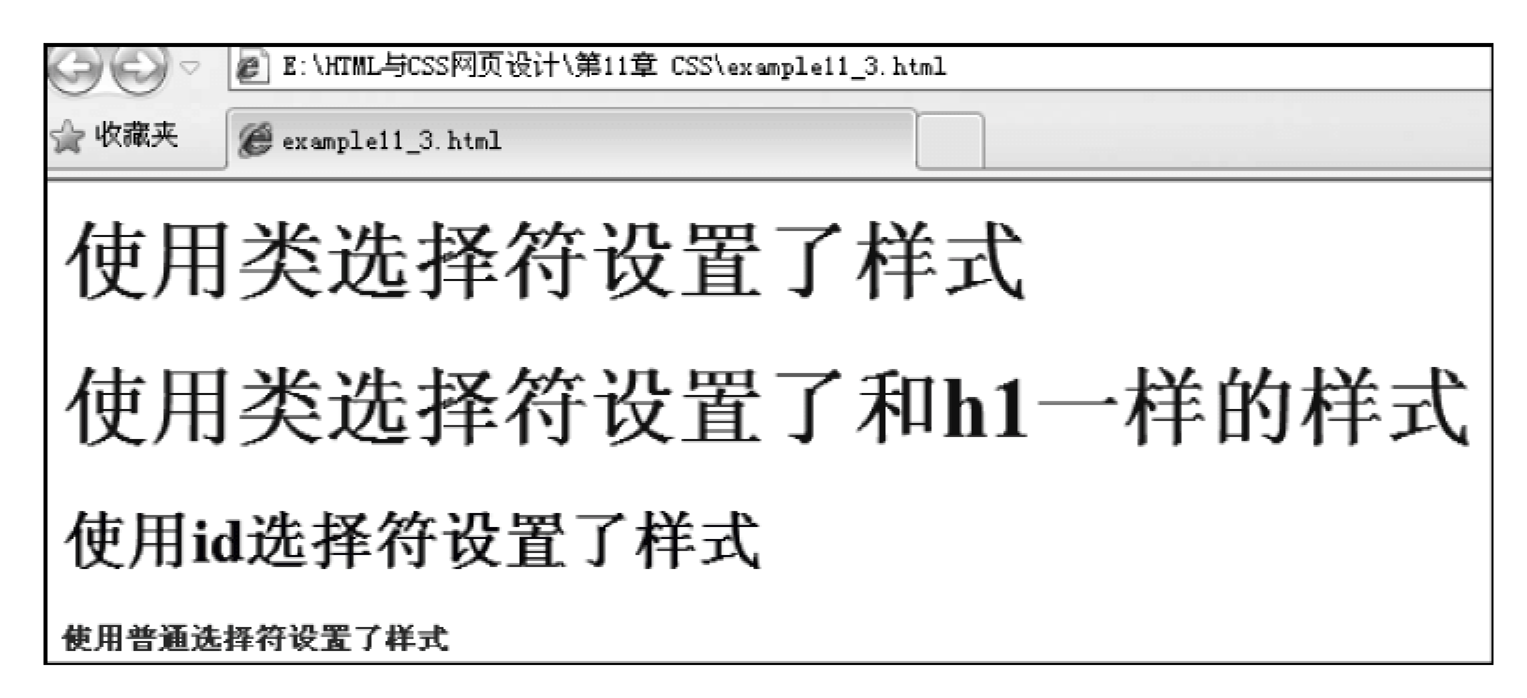


图 11.2 example11_3.html 运行效果

2. 任务的代码模板

将下列 example11_3.html 中的【代码】替换为程序代码。

example11_3.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_3.html</title>
<style type="text/css">
    /* 代码 1 定义名为 h1Andh2Class 的通用类选择符 */
    【代码 1】{
        font-size: 28pt;
```

```
        color: red
    }
    /* 代码 2 定义名为 myId 的通用 id 选择符 */
    【代码 2】{
        font-size: 20pt;
        color: blue
    }
    h4{
        font-size: 10pt;
        color: green
    }
</style>
</head>
<body>
    <!-- 代码 3 引用名为 h1Andh2Class 的类选择符设置样式 -->
    <h1【代码 3】>使用类选择符设置了样式</h1>
    <!-- 代码 4 引用名为 h1Andh2Class 的类选择符设置样式 -->
    <h2【代码 4】>使用类选择符设置了和 h1 一样的样式</h2>
    <!-- 代码 5 引用名为 myId 的 id 选择符设置样式 -->
    <h3【代码 5】>使用 id 选择符设置了样式</h3>
    <h4>使用普通选择符设置了样式</h4>
</body>
</html>
```

3. 任务小结或知识扩展

普通选择符有一定的局限性，它针对某个 HTML 标记有效。而通用的类选择符（省略 HTML 标记）没有 HTML 标记的限制。另外，尽量少用 id 选择符，因为在一个页面中，每个 id 只能使用一次。

4. 代码模板的参考答案

```
【代码 1】: .h1Andh2Class
【代码 2】: #myId
【代码 3】: class="h1Andh2Class"
【代码 4】: class="h1Andh2Class"
【代码 5】: id="myId"
```

11.3.4 实践环节

将 example11_3.html 中的通用类选择符和通用 id 选择符改成类选择符和 id 选择符。

11.4 伪类及伪对象

伪类及伪对象由 CSS 自动支持，属于 CSS 的一种扩展类型。名称不能被用户自定义，使用时只能按照标准格式进行应用。

11.4.1　核心知识

1. 超链接伪类

超链接伪类共有 4 个，a：link、a：visited、a：hover 和 a：active。a：link 表示未被访问的链接，a：visited 表示已被访问过的链接，a：hover 表示鼠标悬浮在上的链接，a：active 表示鼠标单击激活链接。由于优先级的关系，在写超链接<a>的 CSS 时，一定要按照 a：link、a：visited、a：hover、a：active 的顺序书写。如：

```
a:link {color: red}                /* 未被访问的链接 红色 */
a:visited {color: green}           /* 已被访问过的链接 绿色 */
a:hover {color: yellow}            /* 鼠标悬浮在上的链接 黄色 */
a:active {color: blue}             /* 鼠标单击激活链接 蓝色 */
```

2. 常用伪对象

：first-letter 设置对象内的第一个字符的样式表属性，如设置段落 p 标记的第一个字符的样式代码如下：

```
p:first-letter {
    color: red;
    font-size: 16px
}
```

：first-line 设置对象内的第一行的样式表属性，如设置 body 对象里第一行的样式代码如下：

```
body:first-line {
    color: red;
    font-size: 16px
}
```

11.4.2　能力目标

理解并掌握常见伪类及伪对象的使用方法。

11.4.3　任务驱动

1. 任务的主要内容

编写网页 example11_4. html，具体要求如下：

- 页面中有 h1 标题，使用伪对象：first-letter 设置 h1 标题的首字符样式；
- 网页运行效果如图 11.3 所示。

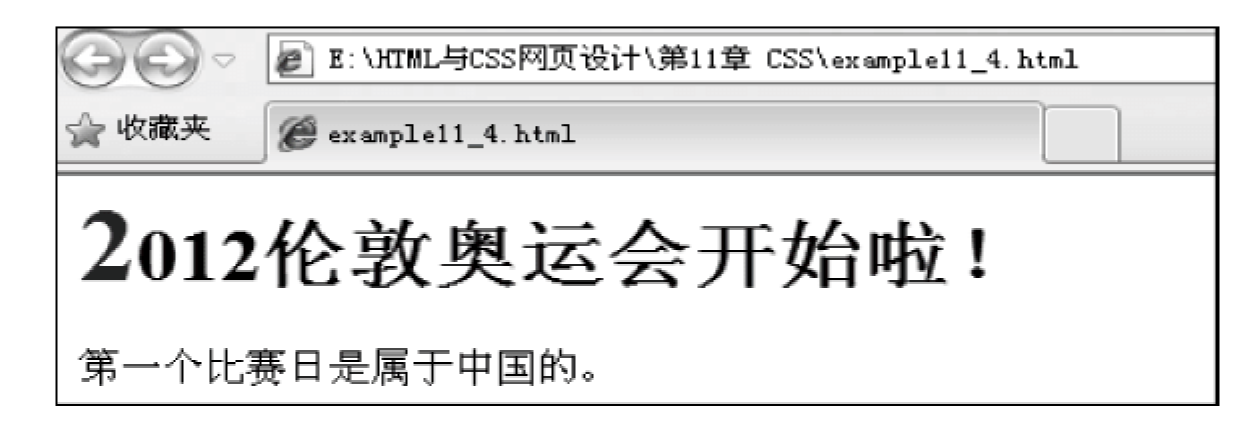


图 11.3　example11_4. html 运行效果

2. 任务的代码模板

将下列 example11_4. html 中的【代码】替换为程序代码。

example11_4. html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_4.html</title>
<style type="text/css">
    /* 代码 1 使用伪对象:first-letter 设置 h1 标题的首字符样式 */
    【代码 1】{
        color: red;
        font-size: 44px
    }
</style>
</head>
<body>
    <h1>2012 伦敦奥运会开始啦!</h1>
    第一个比赛日是属于中国的。
</body>
</html>
```

3. 任务小结或知识扩展

伪类可以看做是一种特殊的类选择符，是能被支持 CSS 的浏览器自动识别的特殊选择符。伪类的语法是在原有的语法里加上一个伪类(pseudo-class)：

```
selector:pseudo-class {property: value} (选择符:伪类{属性: 值})
```

伪类和类不同，是 CSS 已经定义好的，不能像类选择符一样随意用别的名字，根据上面的语法可以解释为对象(选择符)在某个特殊状态下(伪类)的样式。

4. 代码模板的参考答案

```
【代码 1】: . h1:first-letter
```

11.4.4 实践环节

编写一个网页 pratice11_4. html 和一个 CSS 文件 mycss. css，具体要求如下。

- 在网页 pratice11_4. html 中有 4 个超链接；
- 在 CSS 文件 mycss. css 中使用超链接伪类设置超链接的样式：未被访问的链接为红色，已被访问过的链接为绿色，鼠标悬浮在上的链接为黄色以及鼠标单击激活的链接为蓝色；
- 在网页 pratice11_4. html 中利用 link 元素添加外部样式文件 mycss. css；
- 网页运行效果如图 11. 4 所示。

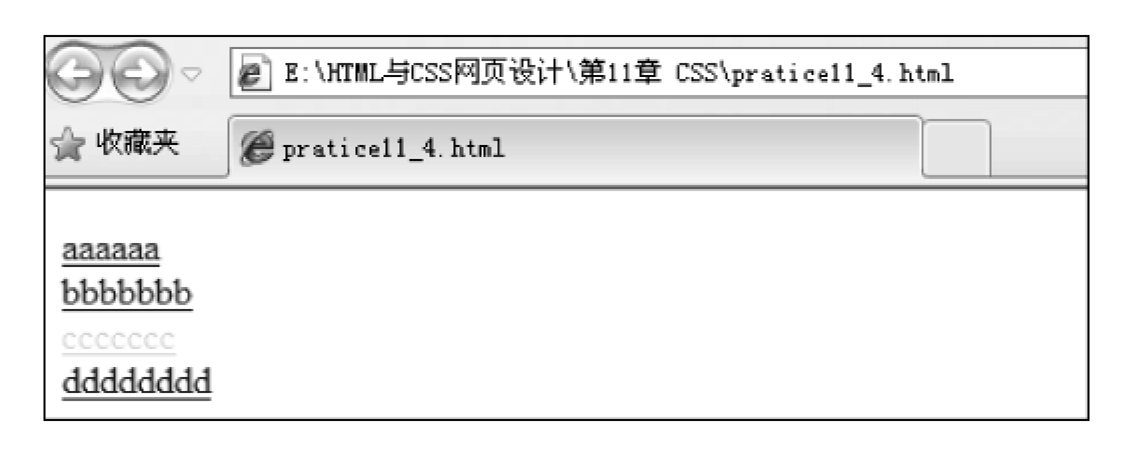


图 11.4　pratice11_4.html 运行效果

11.5　字体属性

CSS 常用字体属性包括字体的类型、大小、颜色等各个方面。

11.5.1　核心知识

1. 字体类型属性(font-family)

该属性设定字体类型，如 Arial、Tahoma、Courier 等。示例代码如下：

```
.s1 {
    font-family: Arial
}
```

2. 字体大小属性(font-size)

该属性设定字体的大小。字体大小的设定可以有多种方式，最常用的是 pt(磅)和 px(像素)。示例代码如下：

```
.s2 {
    font-size: 16pt
}
```

3. 字体风格属性(font-style)

该属性有三个值可选：normal、italic、oblique。normal 是默认值(正常的字体)，italic 和 oblique 都是斜体显示。示例代码如下：

```
.s1 {
    font-sytle: italic
}
```

4. 加粗字体属性(font-weight)

该属性取值包括 normal、bold、bolder 和 lighter。normal 是默认值(正常粗细)；bold 表示粗体；bolder 表示特粗；lighter 表示特细。示例代码如下：

```
<p style="font-weight:bold">
    这段文字的加粗字体属性(font-weight)值是 bold。
<p>
```

5. 字体变体属性(font-variant)

该属性有两个值,normal 和 small-caps,normal 是默认值。small-caps 表示将小写的英文字体转换为大写英文字体。示例代码如下:

```
.s1 {
    font-variant: small-caps
}
```

6. 字体复合属性(font)

该属性可以设定不同字体属性的缩写。可以综合字体风格(font-style)、字体变体(font-variant)、字体加粗(font-weight)、字体大小(font-size)等属性,取值之间用空格区分。示例代码如下:

```
.s1 {
    font: italic bold 11pt arial
}
```

7. 字体颜色属性(color)

该属性可以设定字体的颜色,颜色值可以是英文单词或十六进制值。示例代码如下:

```
.s1 {
    color: red
}
```

11.5.2 能力目标

理解并掌握字体属性的设置方法。

11.5.3 任务驱动

1. 任务的主要内容

编写网页 example11_5. html,具体要求如下:

- 页面中有个 h1 标题“正在学习 CSS 的字体属性”;
- 使用内部样式设置 h1 标题的字体类型为“仿宋_GB2312”,字体风格为斜体,字体颜色为绿色;
- 网页运行效果如图 11.5 所示。



图 11.5　CSS 字体属性

2. 任务的代码模板

将下列 example11_5. html 中的【代码】替换为程序代码。

example11_5. html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_5.html</title>
<style type="text/css">
    h1{
        【代码 1】: 仿宋_GB2312;       /* 代码 1 设置字体类型为仿宋_GB2312 */
        【代码 2】: italic;          /* 代码 2 设置字体风格为斜体 */
        【代码 3】: green            /* 代码 3 设置字体颜色为绿色 */
    }
</style>
</head>
<body>
    <h1>正在学习 CSS 的字体属性</h1>
</body>
</html>
```

3. 任务小结或知识扩展

font-family 属性设置的字体类型受用户浏览器的影响。如果用户浏览器环境中没有相关的字体，则设置的字体将失去作用。

4. 代码模板的参考答案

```
【代码 1】: font-family
【代码 2】: font-style
【代码 3】: color
```

11.5.4 实践环节

将 example11_5. html 中的字体单个属性综合成复合属性(font)。

11.6 文本属性

CSS 常用文本属性包括文本的对齐、缩进、间距等。

11.6.1 核心知识

1. 水平对齐属性(text-align)

该属性取值包括 left(左对齐)、right(右对齐)、center(居中对齐)和 justify(两端对齐)4 种。示例代码如下：

```
#top{
    text-align: center
}
```

2. 文本修饰属性(text-decoration)

该属性可以设定文本的下画线、删除线等。取值有 none(默认值)、underline(对文本添加下画线)、overline(对文本添加上画线)、line-through(对文本添加删除线)以及 blink(闪

烁文本效果）。示例代码如下：

```
#top {
    text-decoration: blink
}
```

3. 文本缩进属性（text-indent）

该属性可以设定文本首行缩进，取值是长度或百分比。长度可以用绝对单位（cm、mm、in、pt、pc）或者相对单位（em、ex、px）。

示例代码如下：

```
#top {
    text-indent: 10px
}
```

4. 文本行高属性（line-height）

该属性可以设定文本中行与行之间的距离，取值是长度或百分比。

示例代码如下：

```
#top {
    line-height: 150%
}
```

5. 大小写转换 text-transform

该属性用来转换英文字母的大小写。取值有 lowercase（使每个单词的首字母大写），uppercase（使每个单词的所有字母大写），capitalize（使每个单词的所有字母小写）。示例代码如下：

```
#top {
    text-transform: lowercase
}
```

6. 单词间隔（word-spacing）

使用 word-spacing 可以设定单词的间隔距离。示例代码如下：

```
#top {
    word-spacing: 1cm
}
```

7. 字符间隔（letter-spacing）

使用 letter-spacing 可以设定字符之间的间隔距离。示例代码如下：

```
#top {
    letter-spacing: 1cm
}
```

11.6.2 能力目标

理解并掌握文本属性的设置方法。

11.6.3　任务驱动

1. 任务的主要内容

编写网页 example11_6. html，具体要求如下：

- 页面中有 3 段话，第一段居左显示，第二段居中显示，第三段居右显示，并且每段首行缩进为 2em；
- 网页最后有个没有下画线的超链接；
- 网页运行效果如图 11. 6 所示。

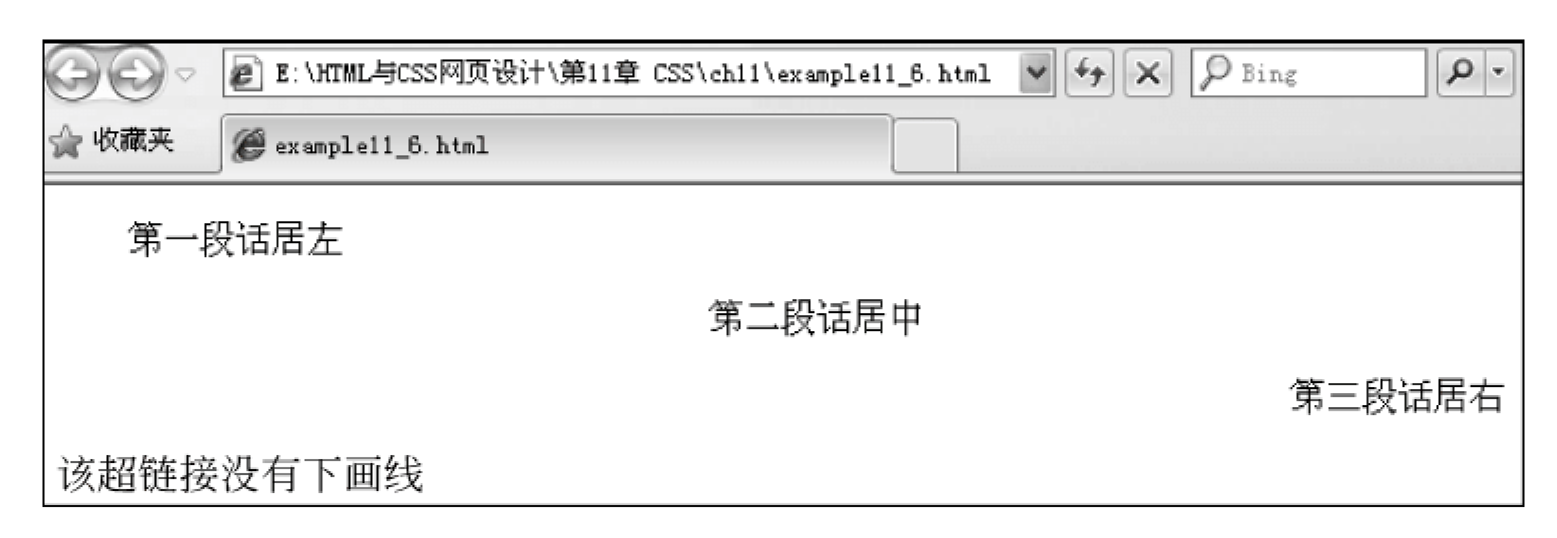


图 11. 6　CSS 文本属性

2. 任务的代码模板

将下列 example11_6. html 中的【代码】替换为程序代码。

example11_6. html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>example11_6.html</title>
    <style type="text/css">
        p{
            【代码 1】/* 代码 1 设置首行缩进为 2em */
        }
        p#first{
            【代码 2】/* 代码 2 设置文本居左对齐 */
        }
        p#second{
            【代码 3】/* 代码 3 设置文本居中对齐 */
        }
        p#third{
            【代码 4】/* 代码 4 设置文本居右对齐 */
        }
        a{
            【代码 5】/* 代码 5 设置文本无下画线 */
        }
    </style>
</head>
<body>
    <p id="first">第一段话居左</p>
    <p id="second">第二段话居中</p>
```

```
    <p id="third">第三段话居右</p>
    <a href="#">该超链接没有下画线</a>
</body>
</html>
```

3. 任务小结或知识扩展

长度单位有相对长度单位和绝对长度单位。

1）相对长度单位

em：相对于当前对象内文本的字体尺寸。

ex：相对于字符“x”的高度，通常为字体高度的一半。

px：像素(Pixel)。

2）绝对长度单位

pt：点(Point)。

pc：派卡(Pica)，相当于我国新四号铅字的尺寸。

in：英寸(Inch)。

cm：厘米(Centimeter)。

mm：毫米(Millimeter)。

1in=2.54cm=25.4mm=72pt=6pc。

4. 代码模板的参考答案

【代码1】：text-indent: 2em
【代码2】：text-align: left
【代码3】：text-align: center
【代码4】：text-align: right
【代码5】：text-decoration: none

11.6.4 实践环节

将example11_6.html页面中所有字符的间距设置为2em。

11.7 背景属性

11.7.1 核心知识

1. 背景颜色(background-color)

该属性为HTML元素设定背景颜色，相当于HTML中的bgcolor属性。示例代码如下：

```
body {
    background-color: green
}
```

2. 背景图片(background-image)

该属性为HTML元素设定背景图片，相当于HTML中的background属性。示例代

码如下：

```
body {
    background-image: url(images/mybg.jpg)
}
```

3. 背景重复（background-repeat）

该属性和 background-image 属性连在一起使用，决定背景图片是否重复。如果只设置 background-image 属性，没有设置 background-repeat 属性，在默认状态下，图片既横向重复，又纵向重复。取值为 repeat-x 时背景图片横向重复；取值为 repeat-y 时背景图片纵向重复；取值为 no-repeat 时背景图片不重复。示例代码如下：

```
body {
    background-image: url(images/mybg.jpg);
    background-repeat: repeat-y
}
```

4. 背景附着（background-attachment）

该属性和 background-image 属性连在一起使用，决定图片是否跟随内容滚动。这个属性有两个值，一个是 scroll（滚动），一个是 fixed（固定）。默认值是 scroll。示例代码如下：

```
body {
    background-image: url(images/mybg.jpg);
    background-repeat: repeat-y;
    background-attachment: fixed
}
```

5. 背景位置（background-position）

该属性和 background-image 属性连在一起使用，决定了背景图片的最初位置。示例代码如下：

```
body {
    background-image: url(images/mybg.jpg);
    background-repeat: repeat-y;
    background-position: 30px 60px
}
```

6. 背景复合（background）

该属性用来综合定义元素的背景，包括背景颜色、背景重复、背景附着和背景位置，之间用空格相连。示例代码如下：

```
body {
    background: green url(images/mybg.jpg) repeat-y 30px 60px
}
```

11.7.2 能力目标

理解并掌握背景属性的设置方法。

11.7.3 任务驱动

1. 任务的主要内容

编写网页 example11_7. html,具体要求如下:

- 页面中有 4 段话,第四段话有背景图片 plane. jpg,并且背景图片不重复;
- 网页运行效果如图 11. 7 所示。



图 11. 7 CSS 背景属性

2. 任务的代码模板

将下列 example11_7. html 中的【代码】替换为程序代码。

example11_7. html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>example11_7.html</title>
<style type="text/css">
    #forth{
        【代码 1】/* 代码 1 设置背景图片为 plane.jpg */
        【代码 2】/* 代码 2 设置背景图片不重复 */
    }
</style>
</head>
<body>
    <p>第一段话</p>
    <p>第二段话</p>
    <p>第三段话</p>
    <p id="forth">第四段话带有背景图片<br><br><br><br>
    <br><br><br><br><br><br><br><br><br>
```

```
    <br><br><br><br><br><br><br><br><br>
    <br><br><br><br><br><br><br><br><br>
    </p>
</body>
</html>
```

3. 任务小结或知识扩展

设置背景图片时，要注意图片文件路径与网页文件路径之间的关系。如果背景图片与背景颜色同时设定，那么背景图片将会覆盖背景颜色。

4. 代码模板的参考答案

【代码 1】: background-image: url(plane.jpg);
【代码 2】: background-repeat: no-repeat

11.7.4　实践环节

将 example11_7.html 中的背景单个属性综合成背景复合属性(background)。

11.8　定位属性

在 CSS 中定位分为绝对定位和相对定位。

11.8.1　核心知识

1. 定位方式(position)

该属性用于对 HTML 标记进行定位。static 为默认值，表示无特殊定位，对象遵循 HTML 定位规则；absolute 表示采用绝对定位，需要同时使用 left、right、top 和 bottom 等属性进行绝对定位，采用绝对定位的对象可层叠；fixed 表示当页面滚动时，元素保持在浏览器视区内；relative 表示相对定位，对象不可层叠，但可根据 left、right、top 和 bottom 等属性设置在页面中的偏移位置。示例代码如下：

```
#top{
    position: absolute;
    top: 0px;
    left:0px;
    right: 100px;
    bottom: 500px
}
```

2. 层叠顺序(z-index)

该属性用于设置层(div)的先后顺序和覆盖关系。其值是无单位的整数值，可以取负值，属性值越高，层就越靠上(可覆盖别的层)。示例代码如下：

```
#top{
    position: absolute;
```

```
    top: 0px;
    left:0px;
    right: 100px;
    bottom: 500px;
    z-index: 2
}
```

3. 浮动(float)

该属性用来改变元素的显示方式，可改变元素的显示位置，同时会对相邻内容造成影响。取值为 left 时，表示文字浮动在元素的右边；取值为 right 时，表示文字浮动在元素的左边。示例代码如下：

```
img{
    float: right
}
```

该示例表示将图像浮动在文字的右边。

4. 宽度(width)和高度(height)

利用 width 和 height 属性可以设置层的宽度和高度。属性值需要设置数值和单位。示例代码如下：

```
#top{
    position: absolute;
    top: 0px;
    left:0px;
    right: 100px;
    bottom: 500px;
    z-index: 2;
    width:300px;
    height: 400px
}
```

11.8.2 能力目标

理解并掌握定位属性的设置方法。

11.8.3 任务驱动

1. 任务的主要内容

编写网页 example11_8.html，具体要求如下：

- 页面中有两个层(DIV)，第一个层里面放的是一张照片(层叠顺序值为 1)，第二个层的层叠顺序值为 2，两个层都使用 CSS 进行绝对定位；
- 网页运行效果如图 11.8 所示。

2. 任务的代码模板

将下列 example11_8.html 中的【代码】替换为程序代码。

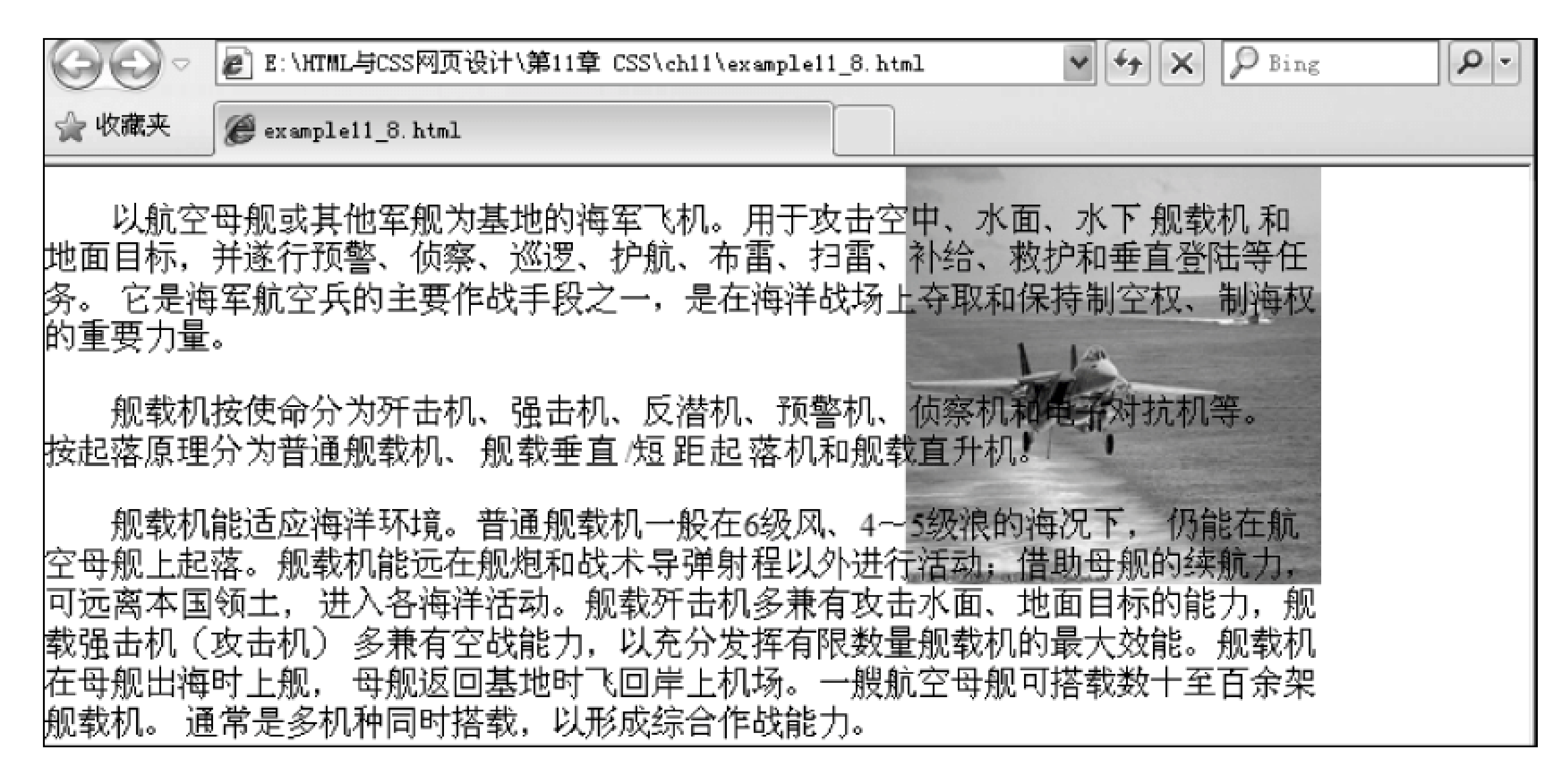


图 11.8　CSS 定位属性

example11_8.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>example11_8.html</title>
    <style type="text/css">
        #div1{
            【代码 1】/* 代码 1 设置绝对定位 */
            【代码 2】/* 代码 2 设置层叠顺序值为 1*/
            top: 0px;
            right: 100px
        }
        #div2{
            【代码 3】/* 代码 3 设置绝对定位 */
            【代码 4】/* 代码 4 设置层叠顺序值为 2*/
            top: 0px;
            left: 0px;
            right: 100px;
        }
    </style>
    </head>
    <body>
        <div id="div1">
            <img src="plane.jpg" width="200" height="200">
        </div>
        <div id="div2">
            <p>以航空母舰或其他军舰为基地的海军飞机。用于攻击空中、水面、水下舰载机和地
               面目标,并遂行预警、侦察、巡逻、护航、布雷、扫雷、补给、救护和垂直登陆等任务。
               它是海军航空兵的主要作战手段之一,是在海洋战场上夺取和保持制空权、制海权
               的重要力量。</p>
            <p>舰载机按使命分为歼击机、强击机、反潜机、预警机、侦察机和电子对抗机等。按起
               落原理分为普通舰载机、舰载垂直/短距起落机和舰载直升机。</p>
            <p>舰载机能适应海洋环境。普通舰载机一般在 6 级风、4~5 级浪的海况下,仍能在航
               空母舰上起落。舰载机能远在舰炮和战术导弹射程以外进行活动;借助母舰的续
```

```
            航力,可远离本国领土,进入各海洋活动。舰载歼击机多兼有攻击水面、地面目标
            的能力,舰载强击机(攻击机)多兼有空战能力,以充分发挥有限数量舰载机的最大
            效能。舰载机在母舰出海时上舰,母舰返回基地时飞回岸上机场。一艘航空母舰
            可搭载数十至百余架舰载机。通常是多机种同时搭载,以形成综合作战能力。
         </p>
      </div>
   </body>
</html>
```

3. 任务小结或知识扩展

在 example11_8.html 页面中 div1 图像层的 z-index 属性为 1,div2 文字层的 z-index 属性为 2。可以看到图像层被文字层覆盖,说明 z-index 属性值越高,它的层越靠上。

4. 代码模板的参考答案

```
【代码 1】: position: absolute;
【代码 2】: z-index: 1;
【代码 3】: position: absolute;
【代码 4】: z-index: 2;
```

11.8.4 实践环节

将 example11_8.html 中两个 div 的 z-index 属性值互换,然后再运行页面,查看运行效果。

11.9 边框属性

11.9.1 核心知识

1. 边框风格(border-style)

该属性用来定义上下左右边框的样式。样式包括无、虚线、实线、双实线等。属性取值为:none(没有边框,无论边框宽度设为多大)、dotted(点线式边框)、dashed(破折线式边框)、solid(直线式边框)、double(双线式边框)、groove(槽线式边框)、ridge(脊线式边框)、inset(内嵌效果的边框)、outset(突起效果的边框)。示例代码如下:

```
.myCss{
    border-style: dotted
}
```

2. 边框宽度(border-width)

该属性用来定义上下左右边框的宽度。属性取值为:medium(默认值)、thin(比 medium 细)、dashed(比 medium 粗)。属性还可以用长度单位定值(绝对长度单位或者相对长度单位)。示例代码如下:

```
.myCss{
    border-width: 20px
}
```

3. 边框颜色(border-color)

该属性用来定义上下左右边框的颜色。示例代码如下：

```
.myCss{
    border-color: cyan
}
```

4. 边框复合(border)

该属性用来定义边框的所有属性,包括风格、宽度和颜色。示例代码如下：

```
.myCss{
    border: cyan 20px dotted
}
```

5. 单边边框

上下左右四个边框不但可以统一设定,也可以分开设定。设定上边框属性,可以使用 border-top、border-top-width、border-top-style 和 border-top-color。设定下边框属性,可以使用 border-bottom、border-bottom-width、border-bottom-style 和 border-bottom-color。设定左边框属性,可以使用 border-left、border-left-width、border-left-style 和 border-left-color。设定右边框,可以使用 border-right、border-right-width、border-right-style 和 border-right-color。

11.9.2　能力目标

理解并掌握边框属性的设置方法。

11.9.3　任务驱动

1. 任务的主要内容

编写网页 example11_9. html,具体要求如下：

- 页面中有两个层(DIV)；
- 第一个层的边框颜色是 green,边框宽度为 2px,边框风格为虚线；
- 第二个层的边框颜色是 blue,边框宽度为 4px,边框风格为双实线；
- 网页运行效果如图 11.9 所示。



图 11.9　CSS 边框属性

2. 任务的代码模板

将下列 example11_9. html 中的【代码】替换为程序代码。

example11_9. html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>example11_9.html</title>
    <style type="text/css">
        #div1{
            position: absolute;
            top: 50px;
            left:50px;
            height: 100px;
            width: 150px;
            【代码 1】/* 代码 1 设置边框的风格为虚线 */
            【代码 2】/* 代码 2 设置边框的颜色为绿色 */
            【代码 3】/* 代码 3 设置边框的宽度为 2px */
        }
        #div2{
            position: absolute;
            top: 50px;
            left:210px;
            height: 100px;
            width: 150px;
            【代码 4】/* 代码 4 设置边框的风格为双实线 */
            【代码 5】/* 代码 5 设置边框的颜色为蓝色 */
            【代码 6】/* 代码 6 设置边框的宽度为 4px */
        }
    </style>
    </head>
    <body>
        <div id="div1">
            第一个层的内容
        </div>
        <div id="div2">
            第二个层的内容
        </div>
    </body>
</html>
```

3. 任务小结或知识扩展

在 CSS 中使用 border 属性设置表格的边框时,仅仅对表格的外边框有效。表格标记 table 有自己的属性 border,border 值大于 0 显示边框,border 值等于 0 隐藏边框。

4. 代码模板的参考答案

```
【代码 1】: border-style: dotted;
【代码 2】: border-color: green;
```

【代码 3】: border-width: 2px
【代码 4】: border-style: double;
【代码 5】: border-color: blue;
【代码 6】: border-width: 4px

11.9.4　实践环节

将 example11_9.html 中的边框单个属性综合成边框复合属性(border)。

11.10　边距与间隙属性

边距属性用来设置页面中一个元素所占空间的边缘到相邻元素之间的距离。间隙属性用来设置边框和元素内容之间的距离。

11.10.1　核心知识

1. 单边距属性(margin-left、margin-right、margin-top 以及 margin-bottom)

单边距属性用来设定元素的某个边距的宽度。示例代码如下:

```
.myClass{
    margin-left: 2cm;
    margin-right: 1cm;
    margin-top: 2cm;
    margin-bottom: 3cm
}
```

2. 复合边距属性(margin)

该属性是设定边距宽度的一个快捷的复合写法,用这个属性可以同时设定上下左右边距属性。可以为上下左右边距设置相同的宽度。示例代码如下:

```
.myClass{
    margin: 1cm;
}
```

也可以分别设置边距,顺序是上、右、下、左。示例代码如下:

```
.myClass{
    margin: 1cm 2cm 3cm 4cm;
}
```

3. 单间隙属性(padding-left、padding-right、padding-top 以及 padding-bottom)

单间隙属性用来设定元素内容到某个边界的宽度。示例代码如下:

```
.myClass{
    padding-left: 2cm;
    padding-right: 1cm;
    padding-top: 2cm;
    padding-bottom: 3cm
}
```

4. 复合间隙属性(padding)

该属性是设定间隙宽度的快捷的复合写法，用这个属性可以同时设定上下左右间隙属性。可以为上下左右间隙设置相同的宽度。示例代码如下：

```
.myClass{
    padding: 1cm;
}
```

也可以分别设置间隙，顺序是上、右、下、左。示例代码如下：

```
.myClass{
    padding: 1cm 2cm 3cm 4cm;
}
```

11.10.2 能力目标

理解并掌握边距与间隙属性的设置方法。

11.10.3 任务驱动

1. 任务的主要内容

编写网页 example11_10. html，具体要求如下：

- 页面中有个两行两列的表格，表格左边距与上边距都为 1cm，单元格的内容上、下、左、右间隙都为 1cm；
- 网页运行效果如图 11. 10 所示。



图 11. 10　CSS 边距与间隙属性

2. 任务的代码模板

将下列 example11_10. html 中的【代码】替换为程序代码。

example11_10. html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>example11_10.html</title>
    <style type="text/css">
```

```
        td{
            【代码 1】/* 代码 1 设置单元格的上间隙为 1cm */
            【代码 2】/* 代码 2 设置单元格的下间隙为 1cm */
            【代码 3】/* 代码 3 设置单元格的左间隙为 1cm */
            【代码 4】/* 代码 4 设置单元格的右间隙为 1cm */
        }
        table{
            【代码 5】/* 代码 5 设置表格的左边距为 1cm */
            【代码 6】/* 代码 6 设置表格的上边距为 1cm */
            border-collapse: collapse /* 相邻边框被合并 */
        }
    </style>
    </head>
    <body>
        <table border=1>
            <tr>
                <td>第一行第一个单元格</td>
                <td>第一行第二个单元格</td>
            </tr>
            <tr>
                <td>第二行第一个单元格</td>
                <td>第二行第二个单元格</td>
            </tr>
        </table>
    </body>
</html>
```

3. 任务小结或知识扩展

边距与间隙属性值可以是百分比，百分比是基于父对象的宽度或高度。

4. 代码模板的参考答案

```
【代码 1】: padding-top: 1cm;
【代码 2】: padding-bottom: 1cm;
【代码 3】: padding-left: 1cm;
【代码 4】: padding-right: 1cm
【代码 5】: margin-left: 1cm;
【代码 6】: margin-top: 1cm;
```

11.10.4 实践环节

将 example11_10.html 中的边距与间隙单个属性综合成复合属性。

11.11 列表属性

11.11.1 核心知识

1. 列表样式类型属性(list-style-type)

该属性用来设定列表项标记的类型。可以取值：disc(默认值，黑圆点)、circle(空心圆

点)、square(小黑方块)、decimal(数字排序)、lower-roman(小写罗马字排序)、upper-roman(大写罗马字排序)、lower-alpha(小写字母排序)、upper-alpha(大写字母排序)、none(无列表项标记)。示例代码如下:

```
.myList{
    list-style-type: decimal
}
```

2. 列表样式位置属性(list-style-position)

该属性用来设定列表的对齐方式。可以取值:outside(以列表项内容为准对齐)和inside(以列表项标记为准对齐)。示例代码如下:

```
.myList{
    list-style-position: inside
}
```

3. 项目符号图像属性(list-style-image)

列表项目符号可以用图片来表示,用项目符号图像属性来设定图片。示例代码如下:

```
.myList{
    list-style-image: url(../images/listImg.gif)
}
```

4. 列表样式复合属性(list-style)

该属性是设定列表样式的快捷的复合写法。用这个属性可以同时设定列表样式类型属性(list-style-type)、列表样式位置属性(list-style-position)和项目符号图像属性(list-style-image)。示例代码如下:

```
.myList {
    list-style: decimal inside url(../images/listImg.gif)
}
```

11.11.2 能力目标

理解并掌握列表属性的设置方法。

11.11.3 任务驱动

1. 任务的主要内容

编写网页 example11_11.html,具体要求如下:

- 页面中有个无序列表,列表项目符号图像为 left.gif,列表样式位置以列表项标记为准对齐;
- 网页运行效果如图 11.11 所示。

2. 任务的代码模板

将下列 example11_11.html 中的【代码】替换为程序代码。

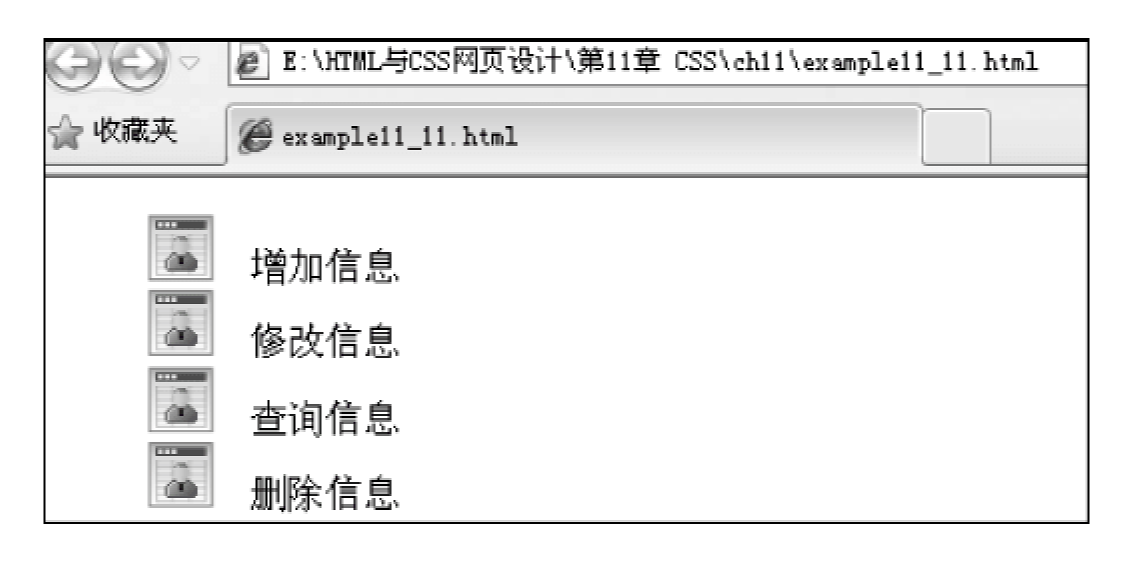


图 11.11　CSS 列表属性

example11_11.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>example11_11.html</title>
    <style type="text/css">
        .myList{
            【代码 1】/* 代码 1 设置项目符号图像为 left.gif */
            【代码 2】/* 代码 2 设置列表样式位置以列表项标记为准对齐 */
        }
    </style>
    </head>
    <body>
        <ul class="myList">
            <li>增加信息</li>
            <li>修改信息</li>
            <li>查询信息</li>
            <li>删除信息</li>
        </ul>
    </body>
</html>
```

3. 任务小结或知识扩展

给列表添加项目符号图像时，要注意图像的尺寸大小适合列表。

4. 代码模板的参考答案

```
【代码 1】: list-style-image: url(left.gif);
【代码 2】: list-style-position: inside
```

11.11.4　实践环节

将 example11_11.html 中的单个列表属性综合成复合列表属性(list-style)。

11.12　常见的 DIV+CSS 布局类型

DIV+CSS 布局是当前网页布局中最流行的类型之一。

11.12.1 核心知识

1. DIV

DIV 是一个放置内容的容器，用于大面积、大区域的块状排版，样式需要编写 CSS 来实现。示例代码如下：

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>div_css1</title>
    <style type="text/css">
    .mainBox {
        border: 1px dashed #0099CC;
        margin: 3px;
        padding: 0px;
        float: left;
        height: 300px;
        width: 192px;
    }
    .mainBox h3 {
        float: left;
        height: 20px;
        width: 179px;
        color: #FFFFFF;
        padding: 6px 3px 3px 10px;
        background-color: #0099CC;
        font-size: 16px;
    }
    .mainBox p {
        line-height: 1.5em;
        text-indent: 2em;
        margin: 35px 5px 5px 5px;
    }
    </style>
    </head>
    <body>
        <div class="mainBox">
            <h3>前言</h3>
            <p>正文内容</p>
        </div>
        <div class="mainBox">
            <h3>CSS 盒子模式</h3>
            <p>正文内容</p>
        </div>
        <div class="mainBox">
            <h3>转变思想</h3>
            <p>正文内容</p>
        </div>
```

```
    </body>
</html>
```

上述代码运行效果如图 11.12 所示。

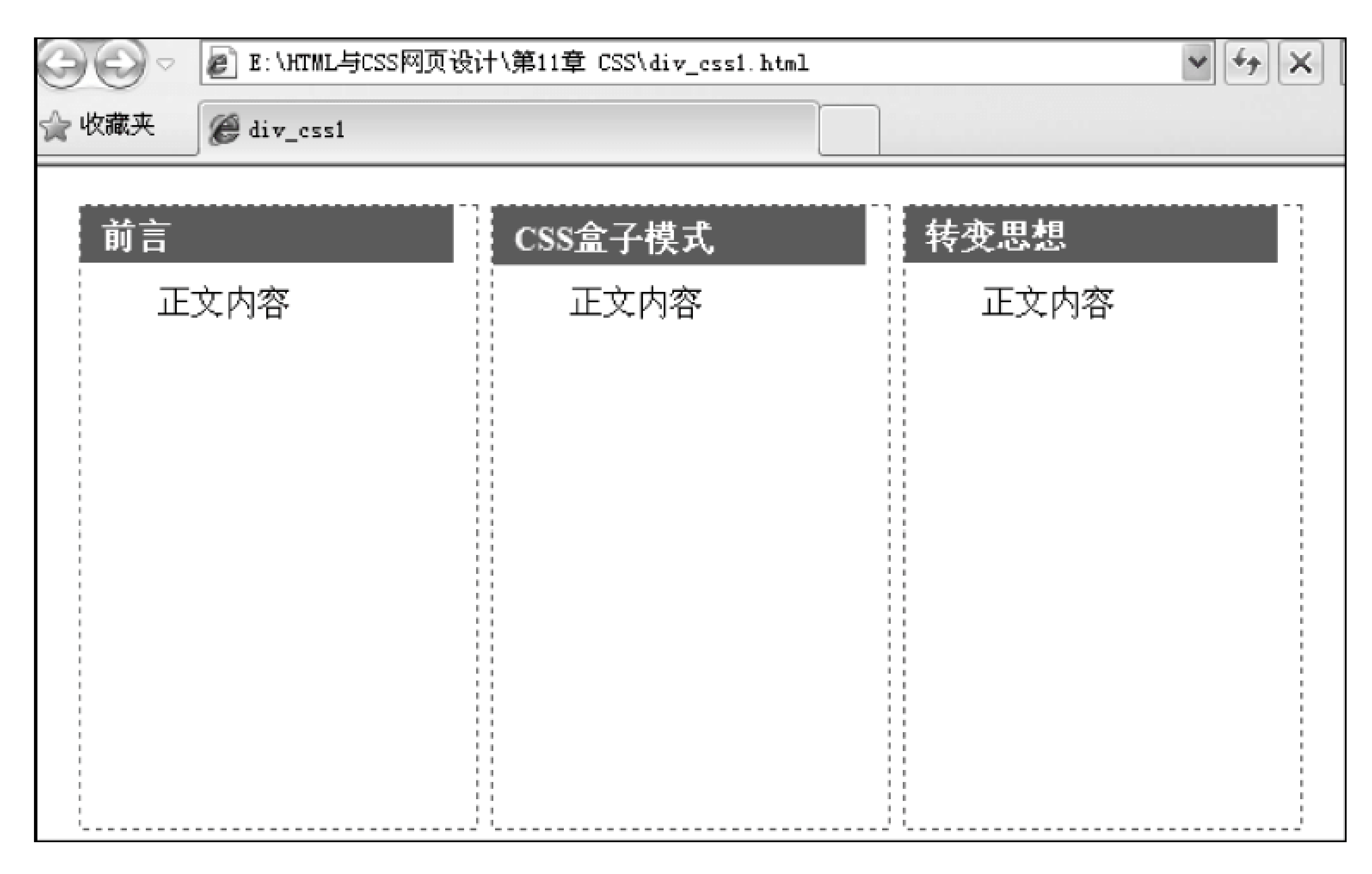


图 11.12　div 示例

2. 一列固定

宽度的属性值是固定像素，无论怎样改变浏览器窗口大小，DIV 的宽度都不改变。示例代码如下：

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>div_css2</title>
    <style type="text/css">
    .oneFixed {
        border: 1px dashed #0099CC;
        background-color: cyan;
        height: 300px;
        width: 300px
    }
    </style>
    </head>
    <body>
        <div class="oneFixed">
            1列固定宽度
        </div>
    </body>
</html>
```

上述代码运行效果如图 11.13 所示。

3. 两列固定宽度

两列的布局需要用到两个 DIV，宽度的属性值是固定像素。示例代码如下：



图 11.13 一列固定宽度

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>div_css3</title>
    <style type="text/css">
    #left {
        border: 1px dashed #0099CC;
        background-color: cyan;
        height: 200px;
        width: 300px;
        float: left
    }
    #right{
        border: 1px dashed #0099CC;
        background-color: LightSkyBlue;
        height: 200px;
        width: 300px;
        float: left
    }
    </style>
    </head>
    <body>
        <div id="left">
            左边
        </div>
        <div id="right">
            右边
        </div>
    </body>
</html>
```

上述代码运行效果如图 11.14 所示。



图 11.14 两列固定宽度

4. 三列浮动中间宽度自适应

三列浮动中间宽度自适应就是要求左边 DIV 固定宽度且左显示，右边 DIV 固定宽度且右显示，中间 DIV 根据左右 DIV 的间距变化宽度自适应。示例代码如下：

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>div_css4</title>
    <style type="text/css">
    body{
        margin: 0px
    }
    #left {
        border: 1px solid #0099CC;
        background-color: cyan;
        height: 300px;
        width: 100px;
        position: absolute;
        top: 0px;
        left: 0px
    }
    #center{
        border: 1px solid #0099CC;
        background-color: #7FFFAA;
        margin-left: 100px;
        margin-right: 100px;
        height: 300px
    }
    #right{
        border: 1px solid #0099CC;
        background-color: cyan;
        height: 300px;
        width: 100px;
        position: absolute;
        right: 0px;
        top: 0px
```

```
        }
        </style>
        </head>
        <body>
            <div id="left">
                左边
            </div>
            <div id="center">
                中间自适应
            </div>
            <div id="right">
                右边
            </div>
        </body>
    </html>
```

上述代码运行效果如图 11.15 所示。



图 11.15　三列浮动中间宽度自适应

5. 三行二列居中高度自适应

三行二列居中高度自适应就是要求整个网页内容居中，第一行 DIV 固定高度且居顶端显示，第三行 DIV 固定高度且居底端显示，中间 DIV 根据内容的变化高度自适应。示例代码如下：

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>div_css5</title>
    <style type="text/css">
    body {
        background: #999;
        text-align: center;
        color: #333;
        font-family: arial, verdana, sans-serif;
    }

    #header {
```

```
    width: 776px;
    margin-right: auto;
    margin-left: auto;
    padding: 0px;
    background: #EEE;
    height: 60px;
    text-align: left;
}

#contain {
    margin-right: auto;
    margin-left: auto;
    width: 776px;
}

#mainbg {
    width: 776px;
    padding: 0px;
    background: #60A179;
    float: left;
}

#right {
    float: right;
    margin: 2px 0px 2px 0px;
    padding: 0px;
    width: 574px;
    background: #ccd2de;
    text-align: left;
}

#left {
    float: left;
    margin: 2px 2px 0px 0px;
    padding: 0px;
    background: #F2F3F7;
    width: 200px;
    text-align: left;
}

#footer {
    clear: both;
    width: 776px;
    margin-right: auto;
    margin-left: auto;
    padding: 0px;
    background: #EEE;
    height: 60px;
}

.text {
```

```
        margin: 0px;
        padding: 20px;
    }
    </style>
    </head>
    <body>
        <div id="header">header</div>
        <div id="contain">
            <div id="mainbg">
                <div id="right">
                    <div class="text">
                        right
                        <p>1</p>
                        <p>1</p>
                        <p>1</p>
                        <p>1</p>
                        <p>1</p>
                    </div>
                </div>
                <div id="left">
                    <div class="text">left</div>
                </div>
            </div>
        </div>
        <div id="footer">footer</div>
    </body>
</html>
```

上述代码运行效果如图 11.16 所示。

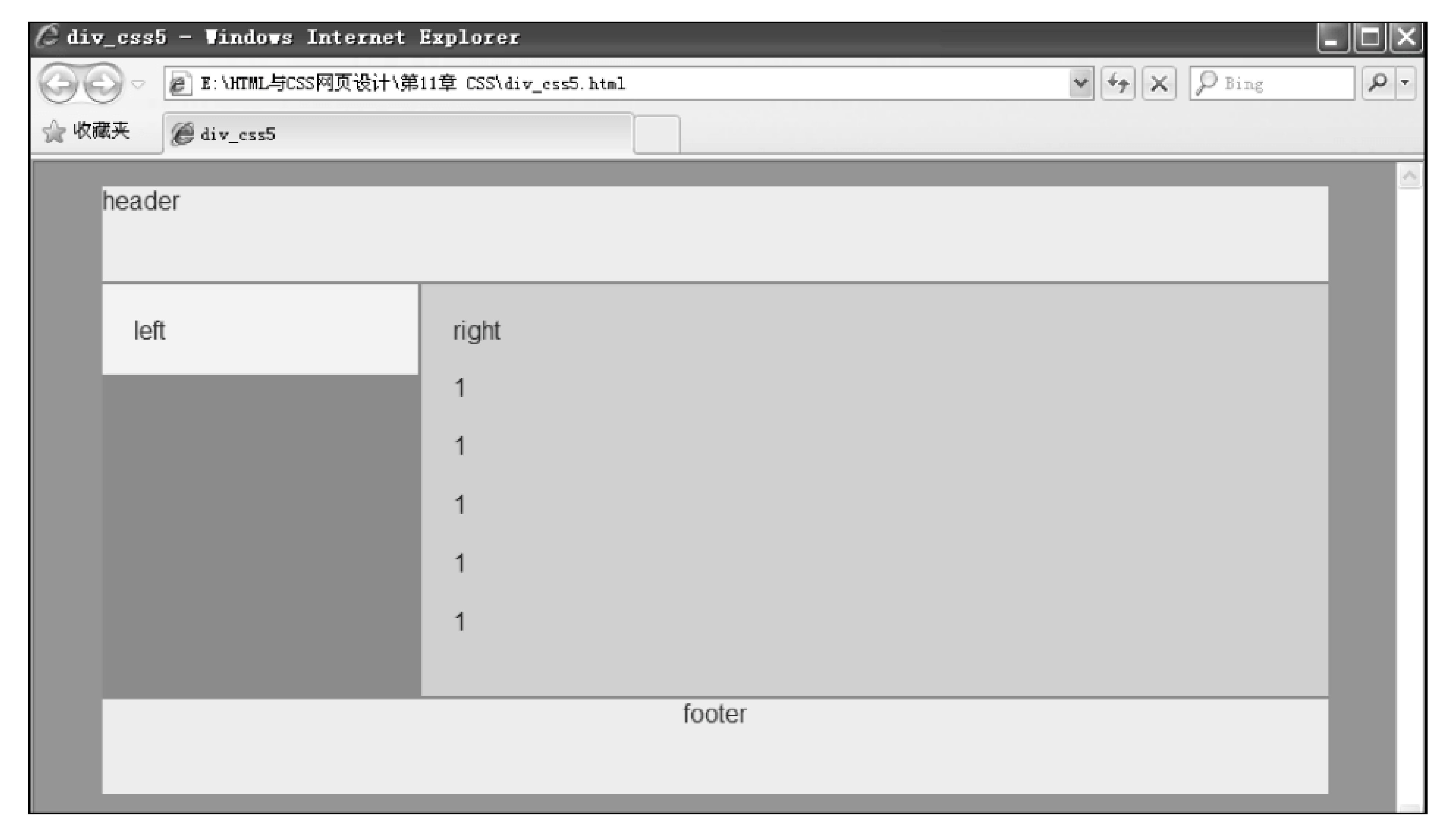


图 11.16　三行二列居中高度自适应

11.12.2　能力目标

理解并掌握常见的 DIV+CSS 布局类型。

11.12.3　任务驱动

1. 任务的主要内容

编写网页 example11_12.html，具体要求如下：

- 页面中有 4 个 div 分别为 top、left、right、bottom，它们都采用绝对定位；
- 在右边(right)div 中有个 iframe；
- 在左边(left)div 中有个 4 行 1 列的表格，每个单元格中有个无下画线的超链接；
- 单击左边 div 中的超链接后，把链接的目标页面显示在右边 div 中的 iframe 里；
- 网页运行效果如图 11.17 所示。

图 11.17　DIV+CSS+iframe 布局

2. 任务的代码模板

将下列 example11_12.html 中的【代码】替换为程序代码。

example11_12.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>信息管理主画面</title>
    <style type="text/css">
    #top {
        【代码 1】/* 代码 1 设置绝对定位 */
        top: 0px;
        left: 200px;
        right: auto;
        bottom: auto;
        background-color: SkyBlue;
        border: 1px solid #ff3399;
        width: 700px;
```

```
    height: 100px;
    text-align: center;
    【代码 2】/* 代码 2 设置字体类型为“华文新魏” */
    【代码 3】/* 代码 3 设置字体大小为 30px */
}

#left {
    float: left;
    position: absolute;
    top: 100px;
    left: 200px;
    right: auto;
    bottom: auto;
    background-color: LightCyan;
    border: 1px solid #9400D3;
    width: 150px;
    height: 400px
}

#right {
    float: right;
    position: absolute;
    top: 100px;
    left: 350px;
    right: auto;
    bottom: auto;
    background-color: #FFB6C1;
    border: 1px solid #EE1289;
    width: 550px;
    height: 400px
}

#bottom {
    position: absolute;
    top: 500px;
    left: 200px;
    right: auto;
    bottom: auto;
    background-color: SkyBlue;
    border: 1px solid #DAA520;
    width: 700px;
    height: 50px;
    text-align: center
}
a{
    【代码 4】/* 代码 4 设置文本无下画线 */
}

</style>
</head>
<body>
```

```
        <div id="top">
            <br>
            欢迎陈恒使用地址簿管理系统
            <br>
        </div>

        <div id="left">
            <table>
                <tr>
                    <td><a href="add.html" target="rightFrame">添加朋友</a></td>
                </tr>
                <tr>
                    <td><a href="update.html" target="rightFrame">修改信息</a></td>
                </tr>

                <tr>
                    <td><a href="delete.html" target="rightFrame">删除朋友</a></td>
                </tr>

                <tr>
                    <td><a href="query.html" target="rightFrame">查询朋友</a></td>
                </tr>

            </table>
        </div>

        <div id="right">
            <iframe src="query.html" name="rightFrame" width="100%" height=
            "100%" frameborder="0"></iframe>
        </div>

        <div id="bottom">
            <br>
            &copy;版权归清华大学出版社所有
            <br>
        </div>
    </body>
</html>
```

add.html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>add.html</title>
    </head>
    <body>
        添加朋友信息
    </body>
</html>
```

update.html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>update.html</title>
    </head>
    <body>
    修改朋友信息
    </body>
</html>
```

delete.html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>delete.html</title>
    </head>
    <body>
        删除朋友信息
    </body>
</html>
```

query.html

```
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>query.html</title>
    </head>
    <body>
        查询朋友信息
    </body>
</html>
```

3. 任务小结或知识扩展

真正使用 div 布局，要把页面的内容和表现完全分离开，在页面的内容部分不应该出现表现控制标记，如 font、color、height、width、align 等标记。

使用 DIV+CSS 布局网页的思路：

- 用 div 来定义结构
- 使用 CSS 来美化网页
- 在 div 定义的区域内加入内容

4. 代码模板的参考答案

【代码 1】: position: absolute;
【代码 2】: font-family: 华文新魏;
【代码 3】: font-size: 30px
【代码 4】: text-decoration: none

11.124 实践环节

按照下面的步骤编写网页 practice11_12.html。

步骤 1：使用 DIV 定义结构。

一个典型的版面分栏结构：页头、导航栏、内容、版权。

结构代码：

```
<div id="header"></div>
<div id="navigator"></div>
<div id="content"></div>
<div id="footer"></div>
```

把这四个盒子装进一个更大的盒子 body 中，如下：

```
<body>
    上面四行代码
</body>
```

步骤 2：定义 body 的属性。

```
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    margin: 0px auto;
    height: auto;
    width: 800px;
    border: 1px solid #006633;
}
```

步骤 3：定义页头(header)的属性。

```
#header {
    height: 100px;
    width: 800px;
    background-image: url(plane.jpg);
    background-repeat: no-repeat;
    margin: 0px 0px 3px 0px;
}
```

步骤 4：定义导航栏(navigator)的属性。

```
#navigator {      /* 定义一个导航栏的长盒子 */
    height: 25px;
    width: 800px;
    font-size: 14px;
    list-style-type: none;      /* 让 nav 这个大盒子下面的小盒子 LI 列表样式不显示,这对于
                                   标准浏览器很重要 */
}

#navigator li {
    float: left;/* 让 LI 这些小盒子左对齐 */
}
```

```
#navigator li a {
    color: #000000;
    text-decoration: none;      /* 让 LI 盒子里面的链接样式没有下画线 */
    padding-top: 4px;
    display: block;     /* 让 LI 里面的链接块状呈现,就像一个按钮,而不必一定要点中链接文字
                           才起作用 */
    width: 131px;
    height: 22px;
    text-align: center;
    background-color: #009966;
    margin-left: 2px;
}

#navigator li a:hover {
    background-color: #006633;      /* 鼠标移到链接盒子上面改变盒子背景色 */
    color: #FFFFFF;
}
```

步骤 5：定义内容部分(content)的属性。

```
#content {
    height: auto;
    width: 780px;
    line-height: 1.5em;
    padding: 10px;
}

#content p {
    text-indent: 2em;
}

#content h3 {
    font-size: 16px;
    margin: 10px;
}
```

步骤 6：定义页脚(footer)的属性。

```
#footer {
    height: 50px;
    width: 780px;
    line-height: 2em;
    text-align: center;
    background-color: #009966;
    padding: 10px;
}
```

步骤 7：定义各标记的边界和填充(开头处)。

```
* {
    margin: 0px;
```

```
    padding: 0px;
}
```

步骤 8：结构代码如下：

```
<body>
    <div id="header"></div>
    <div id="navigator">
        <ul id="navigator">
            <li><a href="#">首页</a></li>
            <li><a href="#">文章</a></li>
            <li><a href="#">相册</a></li>
            <li><a href="#">Blog</a></li>
            <li><a href="#">论坛</a></li>
            <li><a href="#">帮助</a></li>
        </ul>
    </div>
    <div id="content">
        <h3>前言</h3>
        <p>CSS 是英文 Cascading Style Sheet 的缩写,又称为"层叠样式表",简称为样式表。它是
            W3C 定义的标准,一种用来为结构化文档(如 HTML 文档)添加样式(字体、间距和背景等)
            的计算机语言。</p>
        <h3>理解 DIV+CSS 布局</h3>
        <p>简单地说 DIV+CSS(DIV CSS)被称为"WEB 标准"中常用术语之一。首先认识 DIV 是用于
            搭建 html 网页结构(框架)的标签,再认识 CSS 是用于创建网页表现(样式/美化)的样式
            表统称,通过 css 来设置 div 标签样式,这一切我们常常称之为 div+css。
        </p>
    </div>
    <div id="footer">
        <p>关于|广告服务|招聘|客服中心|QQ留言|网站管理|会员登录|购物车</p>
        <p>Copyright (c)清华大学出版社</p>
    </div>
</body>
```

步骤 9：页面运行效果如图 11.18 所示。

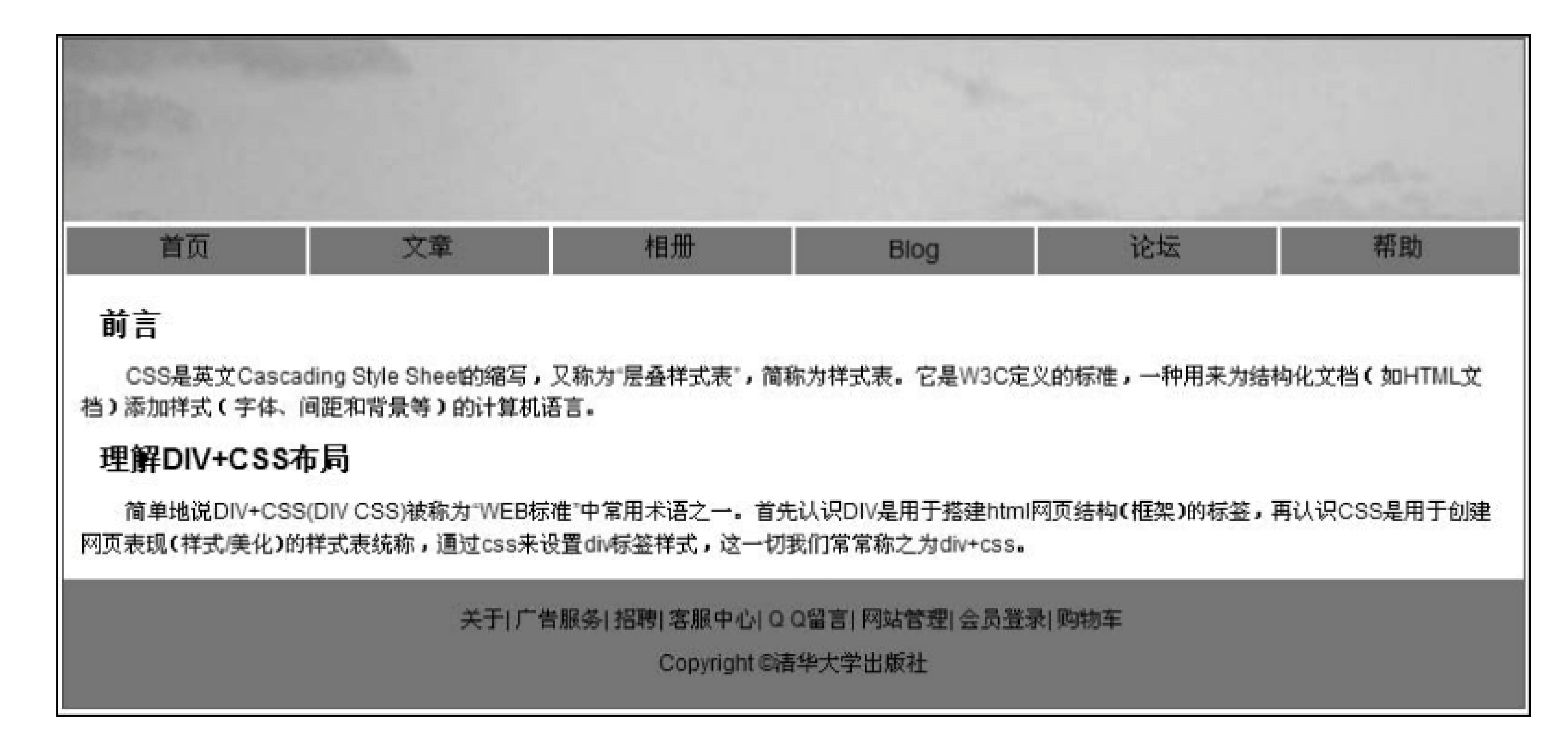


图 11.18　DIV+CSS 布局

11.13 小　　结

CSS的语法由三部分构成：选择符(selector)、属性(property)和属性值(value)。选择符分为：普通选择符、类选择符和id选择符。

在网页中使用CSS有三种方式：内嵌样式、内部样式和外部样式。使用style作为标记的属性进行内嵌样式的设置；在<head>…</head>里面使用<style>标记进行内部样式的设置；通过HTML的link元素进行外部样式的设置。

CSS的常用属性有：字体、文本、背景、定位、边框、边距、间隙以及列表等。

习　题　11

1. 下面对于样式表的说法中错误的是(　　)。
   A. CSS就是Cascading Style Sheets，中文翻译为“层叠样式表”，简称样式表
   B. 将CSS指定的格式加入到HTML中的方法有两种
   C. 外部链接CSS时需要在HTML文件里加一个超级链接，连接到外部的CSS文档
   D. 内定义CSS时需要在HTML文件内的<head>…</head>标签之间加一段CSS的描述内容
2. 下面对于样式表的说法中错误的是(　　)。
   A. CSS的定义由三个部分构成：选择符(selector)、属性(properties)和属性的取值(value)
   B. 选择符可以是多种形式，一般是要定义样式的HTML标记，可以通过此方法定义它的属性和值，属性和值要用逗号隔开
   C. CSS可以定义字体属性
   D. CSS可以定义颜色和背景属性
3. 下列说法中错误的是(　　)。
   A. p {font-family:"sans serif"}定义段落字体为sans serif
   B. p {text-align: center; color: red}定义段落居中排列，并且段落中的文字为红色
   C. background-image属性用来设置背景图片
   D. background-color属性用来设置背景颜色
4. 简述常见的DIV+CSS布局类型。

第 12 章

地址簿网站的布局

本章通过一个小型的地址簿网站，讲述如何采用 DIV+CSS 进行网站的前台布局设计。

12.1 网站功能需求

地址簿网站是针对注册用户使用的。网站提供的功能如下。

（1）非注册用户可以注册为注册用户。

（2）成功注册的用户，可以登录。

（3）成功登录的用户，可以添加、修改、删除以及浏览自己的朋友信息。

（4）成功登录的用户，可以修改自己的登录密码。

12.2 网站内容设计

注册用户使用地址簿网站可以添加、修改、删除以及查询自己的朋友信息，具体的内容如下。

1. 用户注册

新用户填写注册信息，包括用户名、密码、确认密码、身份证号码、性别、住址、邮编、邮箱以及生日等。

2. 用户登录

用户输入用户名、密码进行登录。登录成功，进入网站主页面，包括网页头部、导航栏、内容以及页脚等。

3. 添加朋友信息

用户填写朋友信息表单，包括朋友姓名、生日、电话、E-mail、地址以及关系等信息。提交朋友信息表单时，系统使用 JavaScript 验证信息是否输入以及信息格式是否合法。

4. 修改朋友信息

网站首先根据成功登录的用户名查询出该用户的所有朋友信息，然后用户选择某个朋友进行信息修改。

5. 删除朋友信息

网站首先根据成功登录的用户名查询出该用户的所有朋友信息，然后用户选择某个或多个朋友进行删除。

6. 查询朋友信息

网站根据成功登录的用户名查询出该用户的所有朋友信息。

7. 修改密码

成功登录的用户，从主页面进入该页面修改自己的密码。

8. 退出系统

成功登录的用户，在主页面单击安全退出链接，回到登录页面。

12.3 网站首页的布局设计

根据设计要求，将网站首页(index.html)从上到下分成两个区域 top 和 bottom。首页的布局规划如图 12.1 所示。

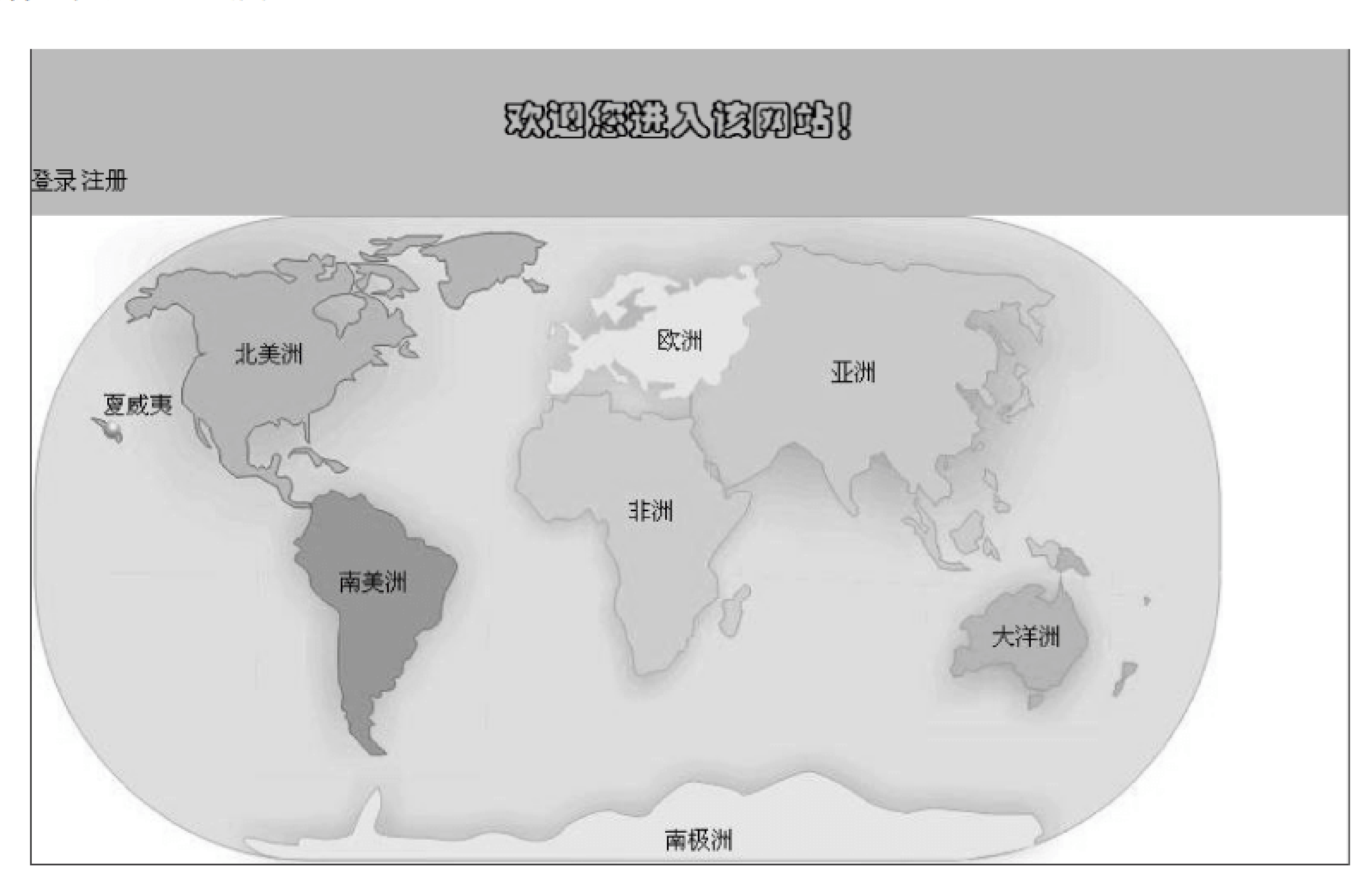


图 12.1 首页的布局规划

网站中用到的 CSS 文件有 common.css、common1.css 和 common2.css。在首页中导入 common.css 文件，首页的代码如下：

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
  loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>index.html</title>
<link href="css/common.css" type="text/css" rel="stylesheet">
<style type="text/css">
#top {
    position: relative;
    top: 0px;
    left: 0px;
```

```
    right: auto;
    bottom: auto;
    background-color: SkyBlue;
    width: 800px;
    height: 100px;
}
#top h1{
    text-align: center;
    font-family: 华文彩云;
    color: #000000;
    font-size: 30px;
}
#top a{
    font-family: Times New Roman;
    text-align: right;
    font-size: 14px;
    color: #000000;
    text-decoration: none;
}
#top a:hover {
    background-color: #006633;
    color: #FFFFFF;
}
#bottom{
    position: relative;
    top: 0px;
    left: 0px;
    right: auto;
    bottom: auto;
    width: 800px;
}
</style>
</head>
<body>
    <div id="top">
        <br><br>
        <h1>欢迎您进入该网站!</h1>
        <br>
        <a href="login.html">登录</a><a href="register.html">注册</a>
    </div>
    <div id="bottom">
        <img alt="" src="images/world1.jpg">
    </div>
</body>
</html>
```

common.css

```
* {
    margin: 0px;
    padding: 0px;
```

```
}
body {
    font-family: Arial, Helvetica,sans-serif;
    font-size: 12px;
    margin: 0px auto;
    height: auto;
    width: 800px;
    border: 1px solid #006633;
}
```

common1.css

```
* {
    margin: 0px;
    padding: 0px;
}
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    margin: 10px 10px auto;
    background-image: url(../images/bb.jpg)
}
```

common2.css

```
* {
    margin: 0px;
    padding: 0px;
}
body {
    font-family: Arial, Helvetica, sans-serif;
    font-size: 12px;
    margin: 0px auto;
    background-image: url(../images/bb.jpg)
}
a {
    color: #000000;
    text-decoration: none;
}
a:hover {
    background-color: #006633;
    color: #FFFFFF;
}
```

12.4 网站主页的布局设计

网站的主页是网站的门面，要设计得简洁、大方，在主页上显示出网站的主要栏目。将主页从上到下分成页头（header）、导航栏（navigator）、内容（content）和版权（footer）四部分，结构代码如下：

```
<body>
    <div id="header"></div>
    <div id="navigator"></div>
    <div id="content"></div>
    <div id="footer"></div>
</body>
```

导航栏和内容的联系：单击导航栏中的链接，把目标页面显示在内容部分中。因此，在内容部分中使用了 iframe，iframe 的默认页面是查询信息页面(query.html)。主页的布局规划如图 12.2 所示。

图 12.2　主页的布局规划

在主页中导入 common.css 文件，主页的代码如下：

main.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
 loose.dtd">
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>main.html</title>
    <link href="css/common.css" type="text/css" rel="stylesheet">
    <style type="text/css">
    #header {
        height: 90px;
        width: 800px;
        background-image: url(images/bb.jpg);
        margin: 0px 0px 3px 0px;
    }
    #header h1{
        text-align: center;
        font-family: 华文彩云;
        color: #000000;
        font-size: 30px;
    }
```

```
#navigator {
    height: 25px;
    width: 800px;
    font-size: 14px;
    list-style-type: none;
}

#navigator li {
    float: left;
}

#navigator li a {
    color: #000000;
    text-decoration: none;
    padding-top: 4px;
    display: block;
    width: 131px;
    height: 22px;
    text-align: center;
    background-color: PaleGreen;
    margin-left: 2px;
}

#navigator li a:hover {
    background-color: #006633;
    color: #FFFFFF;
}
#content {
    height: auto;
    width: 780px;
    padding: 10px;
}
#content iframe{
    height: 300px;
    width: 780px;
}
#footer {
    height: 30px;
    width: 780px;
    line-height: 2em;
    text-align: center;
    background-color: PaleGreen;
    padding: 10px;
}
</style>
</head>
<body>
    <div id="header">
        <br><br>
        <h1>欢迎您进入主页面!</h1>
    </div>
```

```
        <div id="navigator">
            <ul id="navigator">
                <li><a href="query.html" target="center">查询朋友信息</a></li>
                <li><a href="add.html" target="center">添加朋友信息</a></li>
                <li><a href="update.html" target="center">修改朋友信息</a></li>
                <li><a href="delete.html" target="center">删除朋友信息</a></li>
                <li><a href="updatepassword.html">修改个人密码</a></li>
                <li><a href="index.html">安全退出系统</a></li>
            </ul>
        </div>
        <div id="content">
            <iframe src="query.html" name="center" frameborder="0"></iframe>
        </div>
        <div id="footer">Copyright ©清华大学出版社</div>
    </body>
</html>
```

12.5　网站其他页面的布局设计

把首页和主页进行详细的布局设计后，接下来对各个网页进行具体的设计。

1. 登录页面

为了页面的整体美观，登录页面中使用了图像域（“确定”和“取消”按钮）。登录页面的布局规划如图 12.3 所示。



图 12.3　登录页面的布局规划

在登录页面中导入 common1.css 文件，登录页面的代码如下：

login.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
 http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>login.html</title>
    <link href="css/common1.css" type="text/css" rel="stylesheet">
    <style type="text/css">
        table{
            text-align: center
        }
        .textSize{
            width: 120px;
            height: 15px
        }

    </style>
    <script type="text/javascript">
    //确定按钮
    function gogo(){
        document.forms[0].submit();
    }
```

```
    //取消按钮
    function cancel(){
        document.forms[0].action="";
    }
    </script>
    </head>
    <body>
        <form action="main.html" method="post">
        <table>
            <tr>
                <td colspan="2"><img src="images/login.gif"></td>
            </tr>
            <tr>
                <td>姓名:</td>
                <td><input class="textSize" type="text" name="userName"
                value="卡扎菲 2 号"></td>
            </tr>
            <tr>
                <td>密码:</td>
                <td><input class="textSize" type="password" name="pwd"
                value="123456"></td>
            </tr>
            <tr>
                <td colspan="2">
                    <input type="image" src="images/ok.gif" onclick="gogo()">
                    <input type="image" src="images/cancel.gif" onclick="cancel()">
                </td>
            </tr>
        </table>
        </form>
    </body>
</html>
```

2. 注册页面

注册页面包括用户名、密码、确认密码、身份证号码、性别、住址、邮编、邮箱以及生日等注册信息。输入注册信息后，单击“提交”按钮时检验信息是否符合要求。验证规则如下：

- 根据出生年月日判断身份证号码是否合法(只考虑18位身份证，验证日期就可以)；
- 登录账号只能是字母或数字且以字母开头；
- 密码要在8位以上且需要有字母和数字之外的字符；
- 出生年月日的格式为yyyy-mm-dd；
- 邮编为6位数字；
- E-mail的基本格式验证即可。

注册页面的布局规划如图12.4所示。



图12.4 注册页面的布局规划

在注册页面中导入common1.css和wpCalendar.js文件，注册页面的代码如下：

register.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
<html>
    <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>register.html</title>
    <link href="css/common1.css" type="text/css" rel="stylesheet">
    <style type="text/css">
        .textSize{
            width: 120px;
            height: 15px
        }
    </style>
    <script type="text/javascript" src="wpCalendar.js" charset="GBK"></script>
    <script type="text/javascript">
    //去掉前后空格
    function deltrim(x){
        while(x.length>0 && x.charAt(0)==' ')
            x=x.substring(1,x.length);
        while(x.length>0&&x.charAt(x.length-1)==' ')
            x=x.substring(0,x.length-1);
        return x;
    }
    //非空验证
    function isNull(elem,message){
        var va=deltrim(elem.value);
        if(va==""){
            alert(message);
            elem.focus();
            return false;
        }
        return true;
    }
    //验证账号
    function validateId(){
        var first=document.forms[0].id.value.charAt(0);
        var exp=/^[a-z0-9]+$/;
        if(isNull(document.forms[0].id,"请输入账号")){      //验证非空
            //验证首字符
            if(first>='a'&&first<='z'||first>='A'&&first<='Z'){
            }
            else{
                alert("账号首字符必须是字母!");
                document.forms[0].id.focus();
                return false;
            }
            if(!exp.test(document.forms[0].id.value)){
                alert("账号必须是字母或数字!");
                document.forms[0].id.focus();
```

```
                return false;
            }
            return true;
        }else{
            return false;
        }
    }
    //验证密码
    function validatepwd(){
        var exp=/^[a-z0-9]+$/;
        if(isNull(document.forms[0].pwd,"请输入密码")){      //验证非空
            if(document.forms[0].pwd.value.length<=8){
                alert("密码大于 8 位");
                document.forms[0].pwd.focus();
                return false;
            }else{
                if(exp.test(document.forms[0].pwd.value)){
                    alert("密码需要有字母和数字之外的字符!");
                    document.forms[0].pwd.focus();
                    return false;
                }
            }
        }else{
            return false;
        }
        if(document.forms[0].pwd.value!=document.forms[0].repwd.value){
            alert("两次密码不一致!");
            document.forms[0].pwd.focus();
            return false;
        }
        return true;
    }
    //验证邮编
    function checkecode(){
        var exp=/^[0-9]+$/;
        if(isNull(document.forms[0].ecode,"请输入邮编")){    //验证非空
            if(document.forms[0].ecode.value.length!=6){
                alert("邮编 6 位");
                document.forms[0].ecode.focus();
                return false;
            }else{
                if(!exp.test(document.forms[0].ecode.value)){
                    alert("邮编为数字");
                    document.forms[0].ecode.focus();
                    return false;
                }
            }
            return true;
        }else{
            return false;
        }
```

```
}
//验证 Email
function checkemail(){
    var exp=/^\w+((-\w+)|(\.\w+))*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)*\.
            [A-Za-z0-9]+$/;
    if(isNull(document.forms[0].email,"请输入 Email")){        //验证非空
        if(!exp.test(document.forms[0].email.value)){
            alert("Email 格式错误!");
            document.forms[0].email.focus();
            return false;
        }
        return true;
    }else{
        return false;
    }
}
//验证身份证
function checkIdenty(){
    var bir=document.forms[0].birthday.value;
    var ide=document.forms[0].identy.value;
    var biride=ide.substring(6,10)+"-"+ide.substring(10,12)+"-"+ide.substring(12,
               14);
    if(isNull(document.forms[0].identy,"请输入身份证")){      //验证非空
        if(ide.length!=18){
            alert("身份证 18 位");
            document.forms[0].identy.focus();
            return false;
    }else{
            if(bir!=biride){
                alert("身份证输入和生日不符合!");
                document.forms[0].identy.focus();
                return false;
            }
        }
        return true;
    }else{
        return false;
    }
}
//提交按钮
function gogo(){
    if(validateId()&&validatepwd()&&checkIdenty()&&checkecode()&&
    checkEmail()){
        document.forms[0].submit();
        return true;
    }
    return false;
}
</script>
</head>
<body>
```

```
<form action="login.html" method="post">
    <table>
        <tr>
            <td>登录姓名</td>
            <td><input class="textSize" type="text" name="id" />
            </td>
        </tr>
        <tr>
            <td>密码</td>
            <td><input class="textSize" type="password" name="pwd" />
            </td>
        </tr>
        <tr>
            <td>确认密码</td>
            <td><input class="textSize" type="password" name="repwd" />
            </td>
        </tr>
        <tr>
            <td>姓名</td>
            <td><input class="textSize" type="text" name="name" />
            </td>
        </tr>
        <tr>
            <td>身份证号码</td>
            <td><input class="textSize" type="text" name="identy" />
            </td>
        </tr>
        <tr>
            <td>出生年月日</td>
            <td><input class="textSize" type="text" name="birthday" readonly
            onfocus="showCalendar(this)" />
            </td>
        </tr>
        <tr>
            <td>住址</td>
            <td><input class="textSize" type="text" name="address" />
            </td>
        </tr>
        <tr>
            <td>邮编</td>
            <td><input class="textSize" type="text" name="ecode" />
            </td>
        </tr>
        <tr>
            <td>E-mail</td>
            <td><input class="textSize" type="text" name="email" />
            </td>
        </tr>
        <tr>
             <td align="right"><input type="image" src="images/submit.gif"
             onclick="return gogo()" />
```

```
                </td>
                <td><input type="image" src="images/cancel.gif">
                </td>
            </tr>
        </table>
    </form>
  </body>
</html>
```

3. 添加朋友信息页面

朋友信息包括朋友 ID、朋友姓名、生日、电话号码、E-mail、地址以及关系等。添加朋友信息页面的布局规划如图 12. 5 所示。

添加朋友信息	
朋友ID	201208080808888
朋友姓名	*
生日	
电话号码	
E-mail	
地址	
关系	同事
提交	重置

图 12. 5　添加朋友信息页面的布局规划

在添加朋友信息页面中导入 common2. css 和 wpCalendar. js 文件,添加朋友信息页面的代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
 http://www.w3.org/TR/html4/loose.dtd">
  <html>
  <head>
  <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
  <title>add.html</title>
  <link href="css/common2.css" type="text/css" rel="stylesheet">
  <script type="text/javascript" src="wpCalendar.js" charset="GBK"></script>
  </head>
  <body>
  <form action="" method="post">
        <table border=1>
            <caption>
                <font size=4 face=华文新魏>添加朋友信息</font>
            </caption>
            <tr>
                <td>朋友 ID</td>
                <td><input type="text" readonly
                      style="border-width: 1pt;border-style: dashed;border-
                      color: red"name="id" value='201208080808888'></td>
            </tr>
            <tr>
                <td>朋友姓名</td>
                <td><input type="text" name="name" maxlength="20" /><font
                      color="red"> * </font>
                </td>
            </tr>
            <tr>
                <td>生日</td>
                <td><input class="textSize" type="text" name="birthday"
                      readonly onfocus="showCalendar(this)" />
                </td>
            </tr>
```

```
                    <tr>
                        <td>电话号码</td>
                        <td><input type="text" name="telephone" maxlength="20" />
                        </td>
                    </tr>
                    <tr>
                        <td>E-mail</td>
                        <td><input type="text" name="email" maxlength="20" />
                        </td>
                    </tr>
                    <tr>
                        <td>地址</td>
                        <td><input type="text" name="address" maxlength="50" />
                        </td>
                    </tr>
                    <tr>
                        <td>关系</td>
                        <td><select name="relation">
                            <option value="同事">同事</option>
                            <option value="同学">同学</option>
                            <option value="战友">战友</option>
                            <option value="老乡">老乡</option>
                            <option value="亲戚">亲戚</option>
                            <option value="家人">家人</option>
                            <option value="密友">密友</option>
                            <option value="兄弟">兄弟</option>
                            <option value="其他">其他</option>
                        </select></td>
                    </tr>
                    <tr>
                        <td align="center"><input type="button" value="提交" />
                        </td>
                        <td align="left"><input type="reset" value="重置" />
                        </td>
                    </tr>
                </table>
            </form>
        </body>
</html>
```

4. 修改朋友信息页面

修改朋友信息页面的布局规划如图 12.6 所示。



图 12.6　修改朋友信息页面的布局规划

5. 查询朋友信息页面

查询朋友信息页面的布局规划如图 12.7 所示。

6. 删除朋友信息页面

删除朋友信息页面的布局规划如图 12.8 所示。

朋友ID	朋友姓名	关系	查看详情
201208080808081	奥巴马1号	密友	查看详情
201208080808082	奥巴马2号	亲戚	查看详情
201208080808083	奥巴马3号	同事	查看详情
201208080808084	奥巴马4号	一般朋友	查看详情
201208080808085	奥巴马5号	兄弟	查看详情

图 12.7　查询朋友信息页面的布局规划

朋友ID	朋友姓名	关系	查看详情	删除
201208080808081	奥巴马1号	密友	查看详情	删除
201208080808082	奥巴马2号	亲戚	查看详情	删除
201208080808083	奥巴马3号	同事	查看详情	删除
201208080808084	奥巴马4号	一般朋友	查看详情	删除
201208080808085	奥巴马5号	兄弟	查看详情	删除

图 12.8　删除朋友信息页面的布局规划

7. 修改个人密码页面

修改个人密码页面的布局规划如图 12.9 所示。

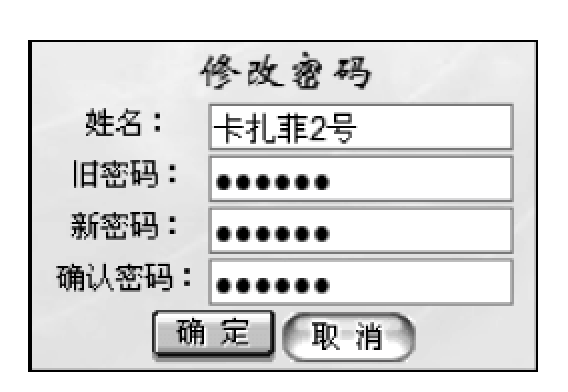


图 12.9　修改个人密码页面的布局规划